TESIS – KS142501

Penerapan Business Intelligence untuk Optimasi Kinerja Aplikasi Perusahaan

HERFIAN SETIAWAN
05211450010021

DOSEN PEMBIMBING
Dr. Sandy Giustiniani
Dr. Jamal Atif
Dr. Apol Pribadi Subriadi, S.T., M.T.

PROGRAM MAGISTER
DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

TESIS – KS142501

Implementation of Business Intelligence for Enterprise Application Performance Optimization

HERFIAN SETIAWAN
05211450010021

SUPERVISORS
Dr. Sandy Giustiniani
Dr. Jamal Atif
Dr. Apol Pribadi Subriadi, S.T., M.T.

MAGISTER PROGRAM
DEPARTMENT OF INFORMATION SYSTEMS
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

l'Implémentation de Business Intelligence pour l'Optimisation des Performances des Logiciels d'Entreprise

Confidentiel Orange

Réalisée par : **Herfian Setiawan**

Responsable Entreprise : **Dr. Sandy Giustiniani**

Responsable Académique : **Dr. Jamal Atif**

Dr. Apol Pribadi Subriadi, S.T., M.T.

Rapport du Stage

Master de MIAGE M2

Université Paris-Dauphine

2015/2017

Penerapan Business Intelligence untuk Optimasi Kinerja Aplikasi Perusahaan

Tesis disusun untuk memenuhi salah satu syarat memeproleh gelar Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember

oleh:
HERFIAN SETIAWAN
NRP. 05211450010021

Tanggal Ujian : September 2016
Periode Wisuda : September 2018

Disetujui Oleh :
Dekan
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Dr. H. Agus Zainal Arifin, S.Kom., M.Kom
NIP. 19720809 199512 1 001

Penerapan Business Intelligence untuk Optimasi Kinerja Aplikasi Perusahaan

Nama Mahasiswa	: Herfian Setiawan
NRP	: 05211450010021
Pembimbing I	: Dr. Sandy Giustiniani
Pembimbing II	: Dr. Jamal Atif
Pembimbing III	: Dr. Apol Pribadi Subriadi, S.T., M.T.

ABSTRAK

Aplikasi atau software sangat diperlukan dalam menjalankan suatu bisnis perusahaan. Semakin maju suatu perusahaan semakin banyak aplikasi yang digunakan dan semakin besar pula data yang tersimpan dalam database perusahaan. Oleh karena itu, aplikasi yang digunakan perusahaan harus diperiksa dan dianalisis setiap tahun guna menghindari menumpuknya aplikasi atau memberikan informasi tentang masih kurangnya aplikasi yang ada.

Penerapan Business Intelligence sebagai teknik untuk mengekstrak data dari berbagai database perusahaan dapat membantu perusahaan lebih mudah mendapatkan data yang relevan sehingga menjadi informasi yang berguna dan mudah dipahami dimana kemudian dilanjutkan dengan analisis statisitik dalam Data Mining.

Metode Data Mining telah banyak diterapkan dalam perusahaan terutama untuk menangani kasus data yang berukuran besar. Dalam masalah klasifikasi, metode Data Mining bekerja dengan sangat baik. Algoritma yang diusulkan untuk mengklasifikasi dan memprediksi kinerja aplikasi perusahaan yaitu Naive Bayes, Decision Tree, Random Forest dan k-Nearest Neighbour.

Dari hasil penelitian diperoleh bahwa metode Decision Tree dengan algoritma J48 adalah metode terbaik dengan nilai akurasi tertinggi dan juga memiliki waktu tercepat dibandingkan dengan metode lainnya yaitu secara berturut-turut 99.92% dan 0.71 detik.

Kata kunci: Data Mining, Business Intelligence, Kinerja Aplikasi, Naive Bayes, Decision Tree, Random Forest, dan k-Nearest Neighbour

Implementation of Business Intelligence for Enterprise Application Performance Optimization

Student's Name	: Herfian Setiawan
Student's ID	: 05211450010021
First supervisor	: Dr. Sandy Giustiniani
Second Supervisor	: Dr. Jamal Atif
Third Supervisor	: Dr. Apol Pribadi Subriadi, S.T., M.T.

ABSTRACT

Application or software is necessary in running a business company. The more advanced a company more and more applications are used and the greater the data stored in the company database. Therefore, the applications that companies use should be checked and analyzed every year to avoid stacking applications or providing information about the lack of existing applications.

Implementation of Business Intelligence as a technique to extract data from various company's databases can help companies more easily to obtain relevant data so that it becomes useful information and easy to understand which then proceed with statistical analysis in Data Mining.

Data Mining methods have been widely applied in companies, especially to handle large data cases. On the classification problem, the Data Mining methods work very well. The proposed algorithms to classify and to predict the performance of enterprise applications are Naive Bayes, Decision Tree, Random Forest and k-Nearest Neighbors.

From the our results, it is found that Decision Tree method with J48 algorithm is the best method with the highest accuracy value and also has the fastest time compared to other methods that is 99.92% and 0.71 seconds respectively.

Keywords: Data Mining, Business Intelligence, Application Performance, Naive Bayes, Decision Tree, Random Forest, and k-Nearest Neighbors

Kata Pengantar

Dengan menyebut nama Allah yang maha pengasih lagi maha penyayang, puji syukur penulis haturkan kepada Allah SWT yang telah memberikan kemampuan pada penulis untuk menyelesaikan tesis yang berjudul "**Penerapan Business Intelligence untuk Optimasi Aplikasi Perusahaan**" sebagai salah satu syarat kelulusan dari Program Pascasarjana Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Tesis ini dilaksanakan dalam program *joint degree* antara Institut Teknologi Sepuluh Nopember (ITS), Indonesia dengan Université Paris-Dauphine, Prancis.

Proses penyusunan tesis ini telah mendapatkan banyak bantuan, bimbingan, masukan serta dukungan dari berbagai pihak. Oleh karena itu, pertama-tama penulis ingin mengucapkan rasa terima kasih kepada Bapak Dr. Jamal Atif selaku kepala departemen MIAGE, yang selalu memotivasi untuk belajar lebih giat dan lebih serius dan juga Bapak Dr. Apol Pribadi Subriadi, S.T., M.T., atas saran bijaknya selama penulis menempuh perkuliaahan di ITS. Selain itu juga kepada Mbak Pian, tata usaha jurusan Sistem Informasi, terima kasih atas bantuan dan kemurahan hatinya.

Kemudian, penulis ingin mengucapkan terima kasih kepada seluruh tim DESI di Perusahaan Orange atas sambutan hangat dan ramah selama enam bulan magang. Lebih khusus lagi, penulis ingin mengucapkan terima kasih kepada tutor magang, Ibu Dr. Sandy Giustiniani atas keramahannya dan bantuannya yang tak ternilai, dan semua kepercayaan yang diberikan kepada penulis selama magang dilakukan.

Selanjutnya, yang paling berharga, penulis berterima kasih kepada orang tua yang selalu memberikan dukungan dan yang tak henti berdoa agar penulis dapat menyelesaikan studi dengan baik. Karena cinta dan kasih sayang dari kedua orang tua, penulis mampu mencapai hingga titik ini.

Terakhir, tentu saja, penulis berterima kasih kepada keluarga besar S2 SI ITS 2014 yang selalu berbagi ilmu dan melewatkan waktu bersama selama menempuh kuliah di ITS.

Paling spesial penulis mengucapkan terima kasih kepada istri tercinta, Nila Novita Gafur, yang selalu mendampingi dalam suka maupun duka.

Daftar Isi

Daftar Figur

Daftar Tabel

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Globalisasi bisnis menciptakan kompetisi yang sangat besar bagi perusahaan. Perusahaan dituntut untuk mampu bersaing secara profesional. Dalam perjalanannya, suatu perusahaan membutuhkan inovasi-inovasi baru guna meningkatkan mutu dari perusahaan tersebut agar dapat bertahan dan berkembang. Pemanfaatan teknologi informasi berperan penting untuk meningkatkan efisiensi dan produktivitas perusahaan. Penerapan teknologi informasi dapat membantu perusahaan dalam mengurangi biaya produksi sehingga perusahaan akan mendapatkan keuntungan yang besar dengan pengeluaran yang kecil. Selain itu, teknologi informasi mempermudah proses komunikasi, akses dan penyebaran informasi, serta memberikan berbagai keuntungan dan kemudahan lainnya dalam menjalankan aktivitas bisnis. Namun, kenyataannya implementasi teknologi informasi ini tidak selalu berjalan efektif meskipun telah mengeluarkan dana investasi yang besar. Misalnya saja, penggunaan aplikasi (*Software*) yang tidak berjalan sesuai harapan. Oleh sebab itu, diperlukannya pengamatan terhadap aplikasi yang akan digunakan dalam perusahaan.

Perusahaan membutuhkan penyusunan data yang akurat agar dapat membantu mengolah data menjadi informasi yang berguna sebagai dasar bagi pengambilan keputusan yang tepat. Data adalah modal mutlak keberhasilan strategi sebuah bisnis. Untuk memenuhi kebutuhan strategi bisnis perusahaan, dapat dilakukan dengan pemanfaatan data-data yang telah ada didalam database perusahaan dengan menggunakan Data Mining [37] . Metode dan algoritma dalam Data Mining membantu *decision maker* memilih dan mengumpulkan data sesuai dengan karakteristiknya masing-masing. Data mining berkerja untuk memberikan pola dan korelasi dari setiap data. Hal ini sangat membantu perusahaan dalam melakukan analisis bisnis dan prediksi.

Perusahaan Orange, salah satu perusahaan telekomunikasi terbesar di Prancis dan beberapa negara di eropa lainnya menyadari betul tentang problematika ini. Organisasi DESI (*Direction de l'Exploitation du SI*) dibawah naungan Orange Company dimobilisasi untuk mendukung proyek-proyek besar. Tujuannya untuk menjamin kualitas kelancaran operasi dan ketersediaan teknologi informasi (TI) yang digunakan karyawan, serta memastikan pengelolaan operasional TI di Grup Orange. DESI mengoperasikan ribuan aplikasi dan server di area grup (konten internet mobile, penawaran TV, pengembangan FTTH, dan portal). Menumpuknya aplikasi yang ada, menuntut organisasi tersebut memilah aplikasi mana yang sudah tidak diperlukan lagi dan memperbaiki aplikasi yang masih memiliki kekurangan tetapi dianggap dapat memberikan manfaat yang besar di masa yang akan datang. Berbagai aplikasi yang tersimpan dalam database ini kemudian digunakan sebagai data untuk kepentingan analisis selanjutnya.

Aplikasi-aplikasi yang digunakan pada perusahaan Orange sangat kompleks seperti Data Mining, Business Object, ekstraksi pengetahuan atau aplikasi web yang menggunakan data yang heterogen dan terdistribusi. Dalam hal ini, kualitas dari keputusan sangat bergantung pada kualitas data yang digunakan. Oleh sebab itu, dibutuhkan teknologi Business Intelligence untuk mengatasi permasalahan tersebut. Untuk menggumpulkan data yang heterogen dan tersebar di banyak database, digunakan proses ETL (Extract, transform, load). Setelah dataset siap, proses prediksi dilakukan dengan beberapa metode klasifikasi yang mana akan dipilih satu metode yang terbaik dan hasil dari prediksi akan dianalisis dari segi performa kinerja aplikasi untuk perusahaan.

Metode Naive Bayes, Decision Tree, Random Forest dan k-Nearest Neighbour diusulkan sebagai metode untuk mengklasifikasi dan memprediksi aplikasi ke dalam beberapa zona atau group. Pengelompokkan tersebut bertujuan untuk melihat kehandalan aplikasi. Selanjutnya prediksi bertujuan untuk mengetahui zona atau group dari aplikasi yang baru. Kemudian, dari keempat metode Data Mining yang diusulkan ini, dilakukan perbandingan berdasarkan nilai akurasi, presisi dan recall serta waktu yang dibutuhkan untuk menjalankan metode-metode tersebut. Dengan demikian, penelitian ini berfokus pada bagaimana menggumpulkan data yang heterogen dan tersebar di banyak database, menstransformasi data, memprediksi dan menganalisis data.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengumpulkan data menggunakan teknologi Business Intelligence?

2. Bagaimana membuat model prediksi terbaik pada data aplikasi?
3. Bagaimana menganalisis hasil prediksi terhadap performa kinerja aplikasi perusahaan?

1.3 Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan yaitu:

1. Mengumpulkan data dengan menggunakan teknologi Business Intelligence.
2. Menerapkan metode Data Mining dalam analisis data aplikasi.
3. Menganalisis perfoma kinerja aplikasi perusahaan terhadap hasil prediksi.

1.4 Ruang Lingkup dan Batasan Masalah

Dalam penelitian ini, masalah yang dibahas akan dibatasi pada hal-hal sebagai berikut:

1. Data aplikasi yang digunakan adalah data rahasia perusahaan telekomunikasi Orange.
2. Algoritma Data Mining yang digunakan adalah Naive Bayes, Decision Tree, Random Forest, dan k-Nearest Neighbour.
3. Data dikumpulkan dari data tahun 2014-2016

1.5 Manfaat Penelitian

1. Hasil dari penelitian ini dapat digunakan perusahaan sebagai bahan pertimbangan dalam mengambil keputusan.
2. Dapat menjadi sumbangsih ilmu pengetahuan khususnya dibidang penerpaan Business Intelligence dan Data Mining.

BAB 2

TINJAUAN PUSTAKA

Pada bab ini, akan diberikan beberapa definisi dan penjelasan teori mengenai metode Data Mining: Naive Bayes, Decision Tree, Random Fores, dan k-Nearest Neighbour serta beberapa teori yang berkaitan dengan penelitian ini yang sebagian besar dapat ditemukan dalam [12], [25] and [43].

2.1 Perangkat Lunak (*Software*)

Perangkat lunak ataupun *software* telah digunakan diberbagai aspek kehidupan manusia, contohnya di perusahaaan, bank, pemerintahan dan masih banyak lagi. Hal ini disebabkan oleh meningkatnya kebutuhan manusia yang diiringi oleh perkembangan teknologi yang pesat. Perangkat lunak adalah sebuah media yang menjalankan hasil dan mengimplementasikan sistem [41]. Perangkat lunak adalah seperangkat instruksi yang dirancang untuk melakukan pemrosesan tertentu pada input ke hasil produk tertentu. Instruksi ini ditulis untuk menangani sistem input-process-output untuk mencapai tujuan yang telah ditentukan [11]. Selain itu, dalam pengertian lain *software* atau perangkat lunak merupakan himpunan dari berbagai instruksi atau perintah yang membuat sebuah komputer dapat bekerja. Instruksi atau perintah ini ditulis oleh seorang *programmer* [15]. Dari beberapa definisi tersebut dapat disimpulkan bahwa *software* itu sendiri secara sederhana memiliki arti sebagai program komputer dimana semua perintah yang digunakan adalah untuk memproses suatu informasi.

Pengertian kinerja atau *performance* merujuk pada tingkat keberhasilan dalam melaksanakan tugas serta kemampuan mencapai tujuan yang telah ditetapkan. Kinerja dinyatakan baik dan sukses jika tujuan yang diinginkan dapat tercapai dengan baik [10]. Kinerja dapat juga diartikan sebagai salah satu kumpulan total dari kerja yang dicapai dan merujuk pada tindakan atau pelaksanaan sesuatu pekerjaan untuk pencapaian tujuan tertentu. Dengan

demikian, kinerja perangkat lunak adalah kinerja aplikasi atau *software* yang bertujuan untuk mencapai hasil yang diinginkan.

2.2 Business Intelligence (BI)

Banyak perusahaan yang sudah memiliki sistem untuk mengumpulkan data dan informasi, tetapi masih sering bingung dalam mengelola data atau tidak memiliki gambaran maupun peta jalan bagaimana mengggunakan data yang ada sebagai informasi yang dapat berguna sebagai pengambilan keputusan yang strategis dan pengembang bisnis [35]. Dengan adanya kemajuan Teknologi Informasi (TI) saat ini, untuk mengumpulkan, menyimpan, memproses, dan membagikan data dapat dilakukan dengan begitu mudah. Hal ini mengakibatkan, data yang tersimpan dalam database misal di suatu perusahaan bergerak meningkat secara perlahan-lahan mengikuti kurva eksponensial. Namun, untuk menjadikan data tersebut menjadi informasi yang dapat diterima oleh banyak orang agar lebih mudah dipahami dalam konteks ini, ada dua istilah yang relevan: Knowledge Discovery (KD) dan Business Intelligence (BI). KD adalah cabang dari bidang Artificial Intelligence (AI) yang bertujuan untuk mengekstraksi pengetahuan tingkat tinggi yang berguna dan dapat dimengerti dari data yang kompleks dan / atau dalam jumlah besar (Fayyad et al., 1996 dalam [7]). BI merupakan istilah umum yang mewakili beberapa arsitektur komputer, alat-alat, teknologi dan metode (misalnya, Data Warehousing, On-line pengolahan analisis dan KD) untuk mengakses data masa lalu dan dukungan pengambilan keputusan di perusahaan publik dan perusahaan, dari operasional untuk tingkat strategis (Turban et al., 2010 dalam [7]).

Dalam literatur lain Business Inteligence (BI) adalah istilah yang umumnya digunakan untuk menggambarkan teknologi, aplikasi, dan proses untuk mengumpulkan, menyimpan, mengakses, dan menganalisis data untuk membantu pengguna membuat keputusan yang lebih baik. Untuk perusahaan berbasis BI, BI adalah sebagai prasyarat untuk bersaing di pasar bisnis [44]. BI bukanlah sebuah produk atau sistem, melainkan sebuah arsitektur dan koleksi operasional yang terintegrasi terhadap aplikasi pengambil keputusan dan database yang menyediakan pelaku bisnis kemudahan akses kepada data bisnis [8].

Dengan demikian dapat pula dikatakan bahwa BI merupakan suatu proses untuk melakukan ekstraksi data-data operasional perusahaan yang selama proses ini dapat juga dilakukan transformasi dengan menerapkan berbagai formula, agregasi, maupun validasi sehingga didapat data yang sesuai dengan kepentingan analisis bisnis [14]. Kemudian data-data tersebut dikumpulkan dalam sebuah data warehouse. Data Warehouse digunakan oleh BI untuk dimining atau diproses dengan menggunakan analisis statistik atau lebih dikenal dengan Data Mining.

Dengan menerapkan BI dalam suatu perusahaan, para pengambil keputusan (*decision maker*) dapat mengintegrasikan sumber data yang berbeda, memprediksi tren, meningkatkan kinerja, melihat indikator kinerja utama, mengidentifikasi peluang bisnis, dan membuat keputusan yang lebih baik dan berdasarkan informasi [2]. Selain itu, dalam [8] manfaat lain dari BI adalah meningkatkan nilai data dan informasi perusahaan, memudahkan pemantauan kinerja perusahaan, meningkatkan nilai investasi teknologi informasi yang sudah ada, menciptakan pegawai yang memiliki akses informasi yang baik (*well-informed workers*), serta yang tidak kalah menarik yaitu dengan BI perusahaan mampu meningkatkan efisiensi biaya.

2.3 Knowledge Discovery in Database (KDD)

Banyak orang yang berpendapat bahwa Data Mining adalah sinonim dari *Knowledge Discovery in Databases* (KDD), kenyataannya Data Mining merupakan salah satu proses dalam KDD seperti yang terlihat pada Gambar 2.1. Karena dalam penelitian ini, data akan diolah dengan tahap-tahap yang ada pada proses KDD berupa *data selection*, *pre-processing*, *data mining*, *post-processing* berupa interpretasi dan evaluasi maka akan dijelaskan lebih rinci tentang tahapan tersebut. Kemudian dilanjutkan dengan penjelasan metode Data Mining serta membahas algoritma yang digunakan untuk menganalisa data.

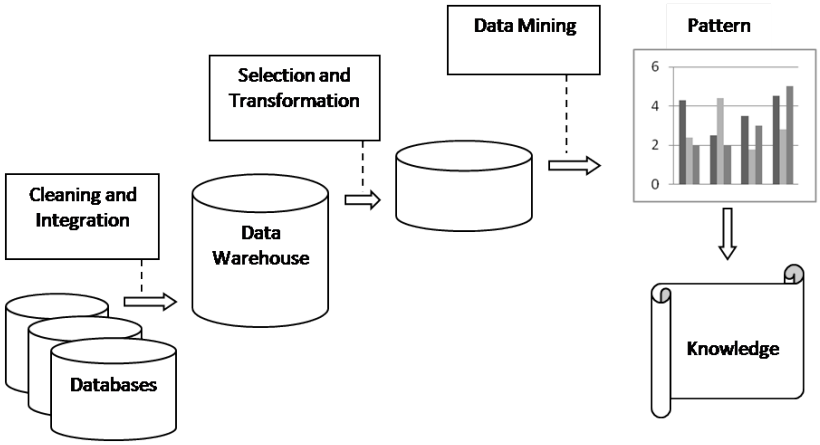

Fig. 2.1 Tahap KDD

Dengan demikian, hanya ada tiga tahap penting dalam KDD yang diterapkan dalam penelitian ini, yaitu:

2.3.1 Preprocessing Data

Sebelum memasuki proses *mining*, ada yang namanya preprocessing data. Dalam tahapan ini, data yang akan diolah harus di *preprocessing* atau pra proses terlebih dahulu karena kebanyakan data asli (*original data*) masih mengandung kerusakan data seperti data error (*noisy data*), data yang tidak lengkap (*incomplete*), tidak konsisten (*inconsistent*), serta ada data yang hilang (*missing value*). Dalam tahapan pra proses ini, masalah-masalah data seperti yang telah disebutkan dapat ditangani. Adapun tahap *preprocessing* data itu sendiri terbagi menjadi beberapa bagian yaitu sbb:

2.3.1.1 Data Cleaning

Data cleaning sesuai dengan arti *clean* yaitu bersih, maka dalam tahap ini data dibersihkan untuk menghilangkan noise data yang tidak konsisten atau kerusakan data yang telah disebutkan sebelumnya. Untuk mengatasi masalah ini dapat dilakukan dengan banyak cara. Contohnya jika data yang missing bersifat numerik maka dapat dilakukan dengan mengisi data yang missing tersebut dengan rata-rata, sementara jika data bersifat nominal maka diisi dengan *modus* (data yang paling banyak muncul). Di sisi lain, dalam hal mengurangi *noisy data*, metode cluster dapat digunakan untuk mengatasinya.

2.3.1.2 Data Integration

Data yang berasal dari berbagai sumber (*source*) yang berbeda akan digabungkan menjadi satu atau menjadi sebuah data yang koheren agar lebih mudah untuk dianalisis seperti dalam data *warehouse*. Dalam *databases* maupun *data warehouses* secara khusus mempunyai metadata yaitu kumpulan aspek data atau sering disebut data tentang data sehingga dapat digunakan untuk membantu menghindari kesalahan dalam skema integrasi. *Data Selection* sering dimasukkan setelah tahap ini, dimana mengambil data yang relevan dengan tugas analisis dari database. Tapi banyak juga para analis yang menyeleksi data terlebih dahulu sebelum melakukan *preprocessing data*.

2.3.1.3 Data Transformation

Data Transformation dilakukan untuk mentransformasi atau menggabungkan data ke dalam bentuk yang sesuai untuk proses *mining*. Dalam tahap ini, *summary* dan normalisasi data sering dilakukan. *Summary* berfungsi untuk membantu melihat kalau data telah sesuai, sedangkan normalisasi berfungsi untuk menyesuaikan skala nilai atribut suatu data sehingga bisa berada pada range tertentu.

2.3.1.4 Data Reduction

Data yang berjumlah besar dan begitu kompleks memerlukan adanya reduksi data sehingga lebih fleksibel dan mudah dianalisis. Data reduksi juga membantu untuk mengurangi pemakaian waktu yang lama dalam analisis data. Dalam data reduksi, data tidak akan kehilangan integritas aslinya, dalam artian masih dapat menghasilkan pengetahuan yang berkualitas. Konsep dari reduksi data yaitu mengecilkan volume atau mengurangi dimensi (jumlah atribut) atau variabel random dari suatu data. Metode yang sering digunakan adalah *Principal Component Analysis* (PCA) dan Random Forest. PCA adalah metode statistik yang sering digunakan untuk menyederhanakan suatu data dimana variabel yang berkorelasi akan ditransformasi ke variabel yang tidak berkorelasi satu sama lain. Sementara Random Forest dibangun dari kumpulan pohon keputusan (Decision Tree) yang didasarkan pada seleksi acak dari data dan variabel. Decision Tree akan dijelaskan pada bagian metode Data Mining.

2.3.2 Data Mining

Setelah data di *preprocessing* mengikuti proses yang telah disebutkan sebelumnya, maka pada tahap ini adalah tahap yang terpenting dimana data akan dianalisis. Pola atau informasi dari data akan diekstrak menggunakan metode-metode cerdas seperti Naive Bayes, Decision Tree, Random Forest, k-Nearest Neighbour dan masih banyak lagi yang akan dibahas lebih lanjut secara terperinci pada bagian selanjutnya.

Jumlah data yang semakin besar dengan pola yang menarik merupakan tantangan bagi manusia untuk mengolah data yang besar tersebut menjadi sebuah informasi dan pengetahuan yang berguna. Tidak hanya data yang besar, sekarang ini data juga memiliki tingkat kompleksitas yang tinggi sehingga pola data menjadi sulit untuk dianalisis dengan cara sederhana atau manual. Pencarian pola data dapat diproses dengan Data Mining. Hal inilah yang mendasari penggunaan Data Mining dalam berbagai dimensi kehidupan baik dalam dunia bisnis, pendidikan maupun dalam pemerintahan. Data Mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, serta *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar [40].

Menurut Garnet Group Data Mining adalah suatu proses untuk menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti statistik dan matematika [25]. Dengan demikian, Data Mining merupakan cara yang digunakan untuk menemukan arti dari pola suatu data dengan menggunakan bantuan komputer guna

1. *Description*: menggambarkan pola dan kecenderungan dalam data atau penggambaran kelompok berdasarkan atribut,
2. *Estimation*: hampir sama dengan klasifikasi hanya saja pada estimasi variable target berbentuk numerik,
3. *Prediction*: hampir sama dengan estimasi dan klasifikasi, namun dalam prediksi nilai dari hasil prediksi akan ada di masa mendatang,
4. *Classification*: berupa pengelompokan yang memiliki kelas tujuan dan bekerja pada data nominal,
5. *Clustering*: berupa pengelompokan data berdasarkan data yang tidak memiliki kelas tujuan atau mengelompokkan objek-objek yang memiliki kemiripan,
6. *Association*: pencarian antar atribut atau menemukan atribut yang muncul dalam satu waktu [24].

Seperti yang terlihat pada Gambar 2.3, Data Mining terbagi menjadi tiga yaitu *Predictive Modeling*, *Discovery* dan *Deviation Detection*. Dalam prakteknya, metode-metode dalam klasifikasi adalah yang paling banyak digunakan untuk mengetahui pola dari suatu data [21] diikuti oleh metode cluster kemudian metode regresi [33] [28].

2.3.3 Postprocessing

Tahapan ini bertujuan untuk menjamin bahwa hasil proses Data Mining yg diintegrasikan pada sistem penunjang keputusan, benar2 hasil yg valid. Tahapan terkahir dari KDD ini berupa intrepretasi hasil dan evaluasi. Interpretasi dilakukan untuk menyimpukan hasil yang diperoleh setelah menerapkan metode-metode Data Mining yang bersesuaian dengan tujuan pembelajaran atau penelitian guna memberikan informasi terkait metode tersebut, sedangkan evaluasi bertujuan untuk memberikan penilaian dan pertimbangan terhadap hasil yang diperoleh sehingga metode terbaik untuk menangani masalah yang ada didapatkan.

Adapun nilai yang biasanya dipertimbangkan sebagai bahan evaluasi kinerja metode untuk mendapatkan metode terbaik adalah nilai akurasi, nilai presisi dan nila recall dari algoritma atau metode tersebut yang semuanya dapat dijelaskan sbb:

2.3.3.1 Akurasi

Pada klasifikasi biner yaitu klasifikasi yang hanya memiliki dua kelas sangat mudah menghitung akurasi dari tipe klasifikasi ini dengan menggunakan persaamaan yang sudah sangat umum yaitu [12]:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

dimana:

- TP : True Positif
- FP : False Positif
- TN : True Negatif
- FN : False Negatif

Untuk dataset yang memiliki label kelas lebih dari dua atau untuk klasifikasi multi kelas, maka agar lebih mudah akurasi dihitung menggunakan Persamaan 2.2.

$$akurasi(y, \hat{y}) = \frac{1}{n_{sampel}} \sum_{i=0}^{n-1} \mathbf{1}_{(\hat{y}_i = yi)} \tag{2.2}$$

dimana y adalah label kelas data aktual, $\hat{y}$ adalah label kelas data prediksi, dan **1** adalah fungsi indikatornya yang dapat ditulikan sbb:

$$1_{(\hat{y}_i = yi)} = \begin{cases} 1 & \text{jika} \hat{y}_i = yi \\ 0 & \text{jika} \hat{y}_i \neq yi \end{cases} .$$

Dari Persamaan 2.2 ini, dapat dilihat bahwa akurasi diperoleh hanya dengan menjumlahkan semua nilai diagonal dari *confusion matrix* dan membaginya dengan banyaknya jumlah observasi yang ada. Selain akurasi, nilai presisi dan recall juga menjadi bahan pertimbangan dalam melakukan klasifikasi dan prediksi [43]. Untuk nilai presisi (*precision*) dihitung dengan menggunakan Persamaan 2.3 atau Persamaan 2.4.

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

Untuk klasifikasi multi kelas agar perhitungannya lebih mudah dimengerti dapat menggunakan rumus:

$$Precision = \frac{M_{ii}}{\sum_i M_{ij}} \tag{2.4}$$

dimana i menandakan baris dan j menandakan kolom dalam confusion matrix, sedangkan recallnya dapat dihitung dengan Persamaan sbb:

$$Recall = \frac{TP}{TP+FN} \tag{2.5}$$

Untuk klasifikasi multi kelasnya dapat menggunakan rumus:

$$Recall = \frac{M_{ii}}{\sum_j M_{ij}} \tag{2.6}$$

2.4 Metode Data Mining

Klasifikasi digunakan untuk mengklasifikasikan setiap item dalam satu set data menjadi satu kelompok kelas. Teknik klasifikasi ini mampu mengolah berbagai data regresi dan semakin populer belakangan ini. Tujuan dari klasifikasi adalah memprediksi secara akurat kelas target untuk setiap kasus dalam data dan algoritmanya bekerja menemukan hubungan antara nilai prediktor dan nilai target. Selain itu, kunci utama masalah dalam hasil klasifikasi adalah mengevaluasi kesalahan dalam pengklasifikasian serta melihat kekuatan hasil prediksi. Prosedur pembuatan model umum untuk pola klasifikasi ditunjukkan seperti pada Gambar 2.4 [36].

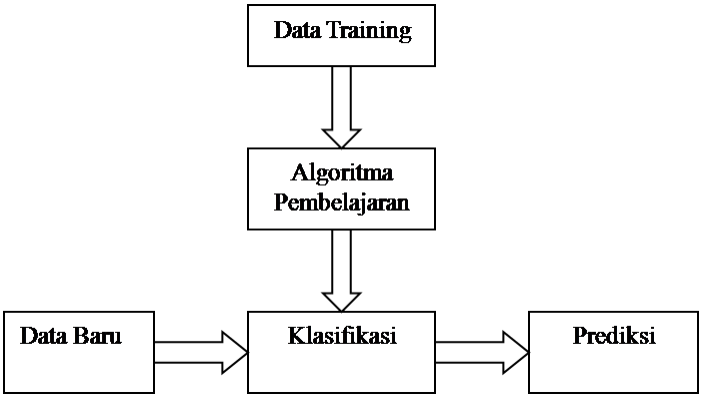

Fig. 2.4 Prosedur Pembuatan Model Klasifikasi

Berikut ini akan dijelaskan metode-metode dari klasifikasi, antara lain:

2.4.1 Naive Bayesian

Metode Naive Bayes adalah metode probabilitas dan statistik yang didasarkan pada teorema Bayes yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa lalu dengan asumsi independen antar variabel (*naive*). Dalam artian tidak ada ketergantungan antara misal ciri tertentu dari sebuah kelas terhadap ciri tertentu dari kelas lain. Metode ini merupakan metode yang paling sederhana diantara semua metode dalam klasifikasi dan

sangat mudah diterapkan karena tidak disertai dengan perkiraan parameter iteratif yang rumit sehingga sangat baik bekerja pada dataset besar [6]. Selain itu, metode ini juga *robust* terhadap *noise* pada data input.

Penerapan metode ini pada data dapat dijelaskan dengan mengambil contoh terdapat data dengan variabel prediktor atau fitur sebanyak n dan anggap ada m kelas dalam variabel respon atau label, maka perhitungan probabilitasnya denggan mengacu pada Teorema Bayes [39] [5] dapat dituliskan sbb:

$$P(C_k|x_1,x_2,\ldots,x_n) = \frac{P(x_1,x_2,\ldots,x_n|C_k).P(C_k)}{P(x_1,x_2,\ldots,x_n)} \tag{2.7}$$

Keterangan:

- C_k mengindikasikan banyaknya kelas dalam variabel respon dimana $k = 1,2,\ldots,m$.
- $P(C_k)$: *class prior probability* atau probabilitas hipotesis kelas target,
- $P(C_k|x_1,x_2,\ldots,x_n)$: *posterior probability* yaitu probabilitas hipotesis kelas target C berdasar kondisi x,
- $P(x_1,x_2,\ldots,x_n|C_k)$: *likelihood* atau probabilitas hipotesis x berdasar pada hipotesis kelas target C,
- $P(x_1,x_2,\ldots,x_n)$: *predictor prior probability* atau probabilitas hipotesis x

Dengan menggunakan asumsi independen naif maka:

$$P(x_i|C_k,x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n) = P(x_i|C_k) \tag{2.8}$$

dan karena $P(x_1,x_2,\ldots,x_n)$ selalu konstan untuk setiap kelas pada satu sampel maka Persamaan 2.7 menjadi:

$$\hat{C} = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{2.9}$$

Metode Naive Bayes dengan persamaan diatas menangani dengan baik hanya pada fitur nominal, tapi tidak bisa secara wajar mengekspresikan kemungkinan antara dua nilai numerik dan pertahankan struktur numerik dari nilai-nilai tersebut. Karenanya untuk data bertipe campuran (*mixed data*), metode Naive Bayes ini dikenal juga dengan *Extended Naive Bayesian classifier* [13]. Untuk fitur numerik, parameter yang diperlukan adalah rata-rata (mean) μ dan varians σ^2 untuk semua kelas yang berbeda C_k. Rata-rata dan varians dapat dicari dengan menggunakan Persamaan 2.10 and Persamaan 2.11 secara berturut-turut:

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\overline{X_i} \tag{2.10}$$

dan variansnya

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \mu\right)^2 \tag{2.11}$$

Perlu diperhatikan bahwa pada setiap fitur numerik, rata-rata dan varians dicari untuk masing-masing kelas, sehingga semua objek yang memiliki kelas yang sama pada satu fitur diperoleh rata-rata dan variannya.

Tahapan-tahapan metode ini dijabarkan dengan jelas pada [13] yaitu sbb:

1. Suatu dataset training membutuhkan pihak-pihak x_i , nilai-nilai kelas C_k dan nilai-nilai atribut w. Jika terjadi perhitungan pada atribut data numerik, hitung nilai rata-rata dan nilai varians menggunakan Persamaan 2.10 and Persamaan 2.11. Namun, waktu dihitung jika atribut data nominal terjadi.

2. Kemudian menghitung probabilitas sebuah objek yang memiliki kelas dan nilai atribut. Jika terjadi perhitungan pada atribut data numerik, probabilitas nilai atribut x_i di kelas C_k menentukan probabilitas menurut persamaan berikut:

$$P(x_i|C_k = P(C_k)\prod_{i=1}^{att_i}P(w_{i,t}|C_k)\,\mathrm{x}\prod_{j=i+1}^{att_j}2P\left(z \geq \frac{|\overline{X_j} - \overline{X_j'}| - (\mu_j - \mu_j')}{\sqrt{\frac{\hat{\sigma_j^2}}{n_j} - \frac{\hat{\sigma_j'^2}}{n_j'}}}\,\mathrm{x}\,q\right) \tag{2.12}$$

dimana z adalah suatu nilai standar (*z-score*) berdasarkan distribusi normal standar.

Jika terjadi perhitungan pada atribut data kategori, probabilitas nilai atribut x_i nilai domain $w_{i,t}$ di kelas C_k menentukan probabilitas seperti:

$$P(C_i|x) > P(C_j|x) \tag{2.13}$$

x berada di kelas C_i; x lainnya berada di kelas C_j.

Pendekatan Bayesian untuk mengklasifikasikan sebuah objek baru adalah menetapkan nilai target yang paling mungkin, $P(C_j|x)$, diberikan nilai atribut $w_1, w_2, \ldots, w_n$ yang menggambarkan objek tersebut sesuai dengan persamaan berikut:

$$P(C_i|x) = \frac{P(C_i \cap x)}{P(x)} = \frac{P(x|C_i)P(C_i)}{P(x)} \tag{2.14}$$

Klasifikasi metode ini membuat asumsi penyederhanaan bahwa nilai atribut secara kondisional independen diberikan nilai target sesuai dengan persamaan berikut:

$$P(x|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{2.15}$$

Nilai-nilai atribut nominal telah dinormalisasi sebelum menghitung probabilitas masing-masing kelas. Ini menentukan normalisasi menurut persamaan berikut:

$$P(w_{i,t}|C_k) = \frac{1 + \sum_{x_i \in C_k} N(w_{i,t}, x_i) P(C_k|x_i)}{|V| + \sum_{x_i \in C_k} \sum_{t=1}^{|V|} N(w_{i,t}, x_i) P(C_k|x_i)} \tag{2.16}$$

di mana $|V|$ adalah nilai total domain dalam nilai atribut x_i.

3. Semua pihak x_i menghitung probabilitas masing-masing kelas, sesuai dengan persamaan berikut:

$$P_i = \prod_{j=1}^{m} P[j] \tag{2.17}$$

4. Kemudian memilih probabilitas maksimal masing-masing kelas sesuai dengan persamaan berikut:

$$\max_{i=1}(P[i]) \tag{2.18}$$

Dengan begitu, objek baru tersebut dikelompokkan pada kelas yang memiliki probabilitas terbesar.

Dalam penelitian terdahulu [18], metode Naive Bayes digunakan untuk memilih grup pelanggan dengan cara mengklasifikasi pelanggan kedalam grup yang sesuai kriteria. Pada penelitian lain [4], klasifikasi dengan metode ini juga diterapkan untuk mengklasifikasi data nasabah asuransi. Penggunaan klasifikasi pada data nasabah ini bertujuan untuk menampilkan informasi klasifikasi lancar, kurang lancar atau tidak lancarnya calon nasabah dalam membayar premi asuransi dengan menggunakan algoritma Naive Bayes.

2.4.2 Decision Tree

Decision Tree atau pohon keputusan adalah representasi grafis dari sebuah prosedur dengan mengklasifikasi atau mengevaluasi item yang diminati sehingga dapat membantu

dalam menghilangkan perhitungan-perhitungan yang tidak diperlukan yang mengakibatkan keputusan yang tadinya kompleks dapat diselesaikan dengan mudah. Pohon keputusan adalah diagram alir yang mirip struktur pohon, dimana setiap simpul menunjukkan tes pada nilai atribut, masing-masing cabang mewakili hasil dari tes, dan daun pohon mewakili kelas atau distribusi kelas [12].

Pohon keputusan dapat digunakan untuk klasifikasi maupun regresi. Pohon keputusan yang memiliki label kelas diskrit disebut *classification tree*, sedangkan yang memiliki label kelas kontinu disebut *regression tree* [42]. Atribut data dalam *classification tree* ini harus berupa data nominal, bila numerik maka atribut harus didiskritkan terlebih dahulu yaitu dengan split dua arah atau biner [43]. Pohon keputusan dapat juga dikatakan sebagai serangkaian keputusan yang berbentuk pohon atau struktur yang berhirarki yang mengarah kepada solusi. Konsep dasar dari metode ini yaitu mengubah data menjadi pohon keputusan dengan aturan-aturan tertentu. Struktur metode ini terdiri dari dua yaitu simpul keputusan dan alternatif. Pada simpul keputusan biasa ditandai dengan tanda kotak, *decision maker* harus memilih satu tindakan alternatif dari jumlah terbatas yang tersedia. Kemudian cabang-cabang yang mengarah kekanan atau kebawah merepresentasikan kumpulan alternatif yang bisa diambil dan ditandai dengan busur. Setiap percabangan menyatakan kondisi yang harus dipenuhi dan tiap ujung pohon menyatakan kelas data. Sebuah alternatif yang tidak dipilih di *prune* atau dipangkas/dihilangkan dan diberi tanda // [45] [1] [17]. Struktur dasar dari metode ini dapat dilihat pada Gambar 2.5.

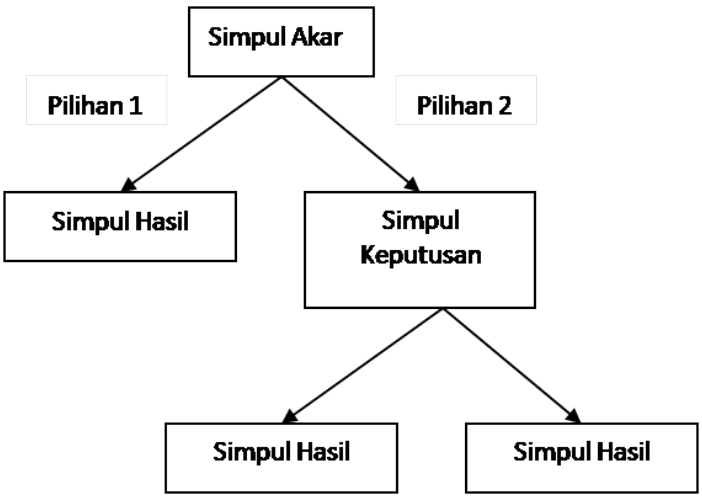

Fig. 2.5 Metode Pohon Keputusan

Metode Decision Tree ini dapat digunakan untuk klasifikasi biner yaitu data yang mempunyai dua kelas (group) dalam labelnya, seperti $Y = [Yes, No]$ atau $Y = [0, 1]$, dst maupun

pada data yang mempunyai banyak kelas (group) dalam labelnya (*multiple class*), dimana label berbentuk $Y = [0,\ldots,\ldots,k-1]$.

Algoritma J48

Algoritma J48 pada Decision Tree merupakan implementasi dari algoritma C4.5 (berbasis Java) pada Weka [23]. Serangkaian perbaikan untuk algoritma ID3 dalam penerapan maupun sistem yang berpengaruh dalam induksi pohon keputusan disebut C4.5. Perbaikan ini termasuk metode untuk menangani data numerik (kontinyu) dan diskrit, dapat menangani nilai atribut yang hilang, data yang noisy, serta menghasilkan aturan-aturan yang mudah diinterpretasikan [43]. Kelebihan lain dari algoritma ini yaitu tercepat diantara algoritma-algoritma yang menggunakan memori utama di komputer [16]. Dengan demikian, menggunakan algoritma J48 di WEKA akan secara otomatis menangani kasus pohon keputusan yang beratribut nominal sekaligus numerik.

Secara umum, tahapan algoritma C4.5 untuk membangun pohon keputusan adalah sbb [31]:

1. Memilih atribut sebagai akar,
2. Membuat cabang untuk masing-masing nilai,
3. Membagi kasus dalam cabang,
4. Mengulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Dalam membangun model berupa pohon keputusan, algoritma C4.5 menggunakan pendekatan teori *information gain* atau *entropy reduction* untuk memilih *split* yang optimal. Dalam memilih atribut untuk memecah obyek dalam beberapa kelas harus dipilih atribut yang menghasilkan *information gain* paling besar [23]. Ukuran *information gain* ini didasarkan pada konsep entropi. Ini biasanya digunakan sebagai ukuran relevansi fitur dalam strategi filter yang mengevaluasi fitur secara individual [32]. Entropi dan informasi gain ini adalah statistik yang membantu dalam membuat pohon keputusan dengan kompleksitas waktu dan ruang yang efisien.

Misal diberikan dataset D dengan $(A_1,A_2,A_3,\ldots,A_n,C)$, $n \geq 1$ dimana C adalah label/atribut kelas. Kemudian diberikan m jumlah dari nilai kelas yang berbeda-beda, entropy dari distribusi kelas dalam D direpresentasikan oleh $entropy(D)$ yang dapat didefinisikan sebagai [32] [23]:

$$entropy(D) = -\sum_{i=1}^{m} p_i * log_2(p_i) \tag{2.19}$$

dimana p_i adalah probabilitas bahwa sebuah objek acak dalam D berada di kelas C_i. Persaamaan ini berlaku ketika data tidak di *split*. Namun, jika dataset di *split* maka untuk menghitung entropy menjadi seperti pada Persamaan 2.20. Misalkan terdapat dataset dengan split S yang mempartisi dataset D menjadi beberapa himpunan bagian (subset) $D_1, D_2, \ldots, D_k$. Kebutuhan informasi rata-rata kemudian dapat dihitung sebagai penjumlahan berbobot dari entropi untuk masing-masing subset sebagai berikut [26]:

$$entropy_S(D) = -\sum_{i=1}^{k} p_i(entropy_S)(D_i) \tag{2.20}$$

dimana *pi* mewakili proporsi tercatat untuk subset *i*.

Kemudian dapat didefinisikan persamaan untuk *information gain* yaitu:

$$Gain(S) = entropy(D) - entropy_S(D) \tag{2.21}$$

peningkatan informasi yang dihasilkan melalui partisi data D sesuai dengan kandidat splis S ini.

Setelah menjelaskan tahapan umum algoritma C4.5, dilanjutkan dengan tahapan yang lebih rinci dari metode pohon keputusan dengan algoritma tersebut [12].

---

1. Membuat simpul akar untuk pohon yang dibuat, N,
2. **Jika** tupel dalam D semuanya mempunyai kelas yang sama, C, **maka**
3. kembalikan N sebagai simpul daun berlabel dengan kelas C tersebut;
4. **If** *atribut list* kosong, **then**
5. kembalikan N sebagai simpul daun yang dilabeli dengan kelas mayoritas dalam D; // jumlah mayoritas
6. Menerapkan **metode pemilihan atribut** (D, *atribut list*) untuk menemukan *kriteria pemisah* "*terbaik*";
7. Label simpul N dengan *kriteria pemisah*;
8. **If** *atribut pemisah* berbentuk diksrit **dan** *multiway* split memungkinkan **then** // tidak terbatas pada pohon biner,

9. *atribut list* ⟵ *atribut list - atribut pemisah* ; // *atribut pemisah*
10. **For each** hasil j dari *kriteria pemisah* // partisi tupel dan membangun subpohon untuk setiap partisi
11. Misal D_j himpunan tupel data dalam D dari output j; // sebuah partisi
12. **If** D_j kosong, **Then**
13. melampirkan daun berlabel dengan kelas mayoritas di D ke node N;
14. **else** lampirkan simpul yang dikembalikan ke (tahapan dalam algoritma C4.5 yang telah disebutkan sebelumnya) (D, *atribut list*) ke simpul N;
15. **Endfor**
16. Kembali ke N;

Keterangan:

- D sebagai data partisi yang semuanya mengacu pada label kelas,
- *Atribut list* merupakan daftar dari atribut yang mendeskripsikan tupel,
- *Metode pemilihan atribut* menspesifikasi prosedur heuristik untuk memilih atribut yang "terbaik" membedakan tupel yang diberikan sesuai dengan kelas. Prosedur ini menggunakan ukuran seleksi atribut seperti *information gain*.
- Tahapan algoritma ini memanggil *metode pemilihan atribut* untuk menentukan *kriteria pemisah*. Kriteria pemisah memberitahu atribut mana yang diuji di simpul N dengan menentukan cara "terbaik" untuk membagi tupel di D ke dalam kelas individu.

Adapun yang berhubungan dengan keutamaan pohon keputusan untuk klasifikasi maupun prediksi tercantum dalam [38]. Dalam penelitiannya, metode ini mudah menangani *missing value* tanpa perlu melakukan imputasi, robust terhadap data *outlier*, dan merupakan pendekatan non-*parametric* tanpa memerlukan asumsi dari distribusi data. Sedangkan kelemahan utama metode ini adalah dapat dikenakan *overfitting* dan *underfitting*, terutama bila menggunakan kumpulan data kecil.

2.4.3 Random Forest atau Hutan Acak

Menurut Breiman, Random Forest adalah kombinasi fitur-fitur pohon sehingga setiap pohon bergantung pada nilai-nilai vektor acak yang diambil secara independen dan dengan distribusi yang sama untuk semua pohon (*trees*) di dalam hutan (*forest*) [3]. Proses pengambilan acak ini disebut juga *boostrap aggregating*. Tujuannya adalah untuk memperkecil varians dan bias dalam klasifikasi yang dibuat oleh Random Forest.

Random Forest tidak cenderung terlalu padat kecuali jika datanya terlalu banyak mengandung noise, jadi memilih banyak pohon keputusan tidak akan menurukan akurasi dari hasil prediksi pada metode ini [27]. Akan tetapi, perlu diperhatikan bahwa semakin banyak algoritma pohon keputusan yang dipakai maka akan semakin banyak pula waktu yang dibutuhkan untuk mengeksekusi program. Metode ini digunakan ketika akurasimenjadi titik acuan dalam sebuah penelitian. Keakuratan Randon Forest tergantung pada kekuatan masing-masing klasifikasi dan ukuran ketergantungan di antara mereka. Idealnya adalah mempertahankan kekuatan pengklasifikasian individu tanpa meningkatkan korelasinya. Randon Forest tidak peka terhadap jumlah atribut yang dipilih untuk dipertimbangkan pada setiap *split* [12].

Cara kerja Random Forest pada prinsipnya sama dengan Decision Tree. Yang membedakannya adalah Random Forest tidak memilih semua titik data dan fitur dalam setiap pohon, tetapi hanya mempertimbangkan subset dari keseluruhan atribut. Algoritma Random Forest menggunakan pendekatan "*catching*" untuk membuat kelompok pohon keputusan dengan subkumpulan data acak. Setelah pohon dibuat, hasilnya digabungkan untuk membuat prediksi akhir. Pembangunan pohon akan berhenti ketika data sudah homogen atau batas jumlah data minimum terlewati. Prediksi akhir dari algoritma Random Forest diperoleh dengan memilah hasil dari semua pohon keputusan atau hanya dengan prediksi yang banyak muncul disetiap kali pohon keputusan dibuat [19] [34].

2.4.4 k-Nearest Neighbour

Algoritma k-*nearest neighbor* (k-NN) adalah sebuah metode sederhana untuk melakukan klasifikasi terhadap objek atau terhadap sekumpulan data berdasarkan data pembelajaran yang paling dekat jaraknya dengan objek tersebut [9]. Klasifiksi terhadap objek baru yang belum diketahui label kelasnya, dapat dilakukan dengan cara sederhana yaitu dengan membandingkan objek tersebut terhadap objek yang paling mirip dalam data *training*. Ukuran kemiripan objek ini dapat ditemukan dengan mengacu pada jarak metrik dengan ketentuan, untuk setiap koordinat x, y, z [25]:

1. $d(x,y) \leq 0$ dan $d(x,y) = 0$ jika dan hanya jika $x = y$

2. $d(x,y) = d(y,x)$

3. $d(x,z) \leq d(x,y) + d(y,z)$

Cara kerja metode ini dapat dipahami, misal ada objek baru x_0 yang ingin diketahui kelas targetnya, maka caranya adalah dengan menghitung jarak objek x_0 terhadap k tetangga terdekat dari objek tersebut. Kemudian mengurutkan semua objek dari yang memiliki jarak terkecil. Hasil klasifikasi didapat dengan melihat kelas mayoritas dari k tetangga terdekat. Jarak pada kNN ini dapat dihitung dengan menggunakan jarak Euclidean [6] (lihat Persamaan 2.22).

$$\begin{aligned} d(x,y) &= \|x-y\| \\ &= \sqrt{\sum_i (x_i - y_i)^2} \end{aligned} \tag{2.22}$$

Persamaan 2.22 berlaku jika atribut berbentuk numerik tapi jika atribut bertipe nominal, maka formula yang dapat digunakan untuk menghitung jarak antar objek adalah dengan jarak Hamming. Jarak Hamming adalah jarak yang dihitung dengan melihat perbedaan antara string bit atau jumlah bit yang berbeda antara dua codeword [12] [43]. Untuk menghitung jarak Hamming dari dua objek atau "string". Misal $u = 1111111$ dan $v = 0000111$, maka jarak Hamming:

$$d(1111111, 0000111) = 4$$

Hal yang sama berlaku untuk kelas yang berbentuk nominal yaitu harus diubah ke bentuk *"string value"* terlebih dahulu.

Pada k-NN umumnya, data harus distandarisasi agar hasil klasifikasi lebih baik terutama bila data memiliki skala yang berbeda ataupun tipe yang berbeda (contoh *mixed data*). Ada beberapa cara untuk menstandarisasi data salah satunya dengan cara *min-max normalization*. Misal untuk mengubah nilai atribut x dari atribut numerik V ke x' dalam range $[0,1]$ yaitu dengan menghitungnya sbb [12]:

$$x' = \frac{x - min_V}{max_V - min_V} \tag{2.23}$$

dimana min_V dan max_V adalah nilai minimum dan maksimum dari atribut V.

Kelebihan dari metode ini yaitu bersifat nonlinier sehingga garis keputusan kelas yang dihasilkan model menjadi begitu fleksibel dan mudah dipahami. Namun, metode ini tidak

menangani data *missing value* secara implisit. Disamping itu, perlu menentukan nilai k optimal atau jumlah tetangga terdekat tanpa adanya aturan baku.

Penggunaan metode kNN untuk klasifikasipun pernah dilakukan sebelumnya seperti pada penelitian [22] dan [20]. Dalam [22], penelitian dilakukan terhadap data pasien yang ada di rumah sakit. Peneliti melihat bahwa begitu banyaknya data pasien masa lampau yang dapat membantu dokter untuk mendiagnosa penyakit pasien dengan lebih mudah. Oleh karena metode kNN mengklasifikasi objek berdasar tetangga terdekat maka diterapkanlah metode ini untuk mengidentifikasi penyakit pasien baru.

BAB 3

METODOLOGI

Pada bab ini akan dijelaskan tentang metodologi penelitian yang meliputi sumber data sera tahap-tahap analisis.

3.1 Sumber Data

Data yang digunakan pada penelitian ini yaitu data yang berkaitan dengan aplikasi perusahaan yang dikumpulan dari berbagai server yang ada di dalam database perusahaan seperti SAP BO, SQL Server, file mentah dan lain-lain. Data aplikasi perusahan ini terdiri dari nama-nama aplikasi yang digunakan perusahaan serta indikator-indikator yang dinilai dari aplikasi perusahaan tersebut mulai dari tahun 2014-2016.

3.2 Variabel Penelitian

Data didapatkan dengan mengumpulkan data yang tersebar diberbagai server, database dan data warehouse, aplikasi yang digunakan adalah SQL server 2012. Dengan bantuan aplikasi tersebut, data dapat diekstrak, ditransformasi dan digabungkan sehingga dapat dianalisis pada tahap selanjutnya.

Dari penggabungan data 2014-2016, yang awalnya terdiri dari jutaan baris diperoleh data setelah digabungkan dan dinormalisasi menjadi 2755 observasi. Adapun yang menjadi indikator-indikator penilaian aplikasi yang ditekankan oleh perusahaan yaitu jumlah insiden yang terjadi di setiap aplikasi, penyebab insiden aplikasi, biaya yang ditimbulkan pada setiap aplikasi, SLA (*service level agreement*) dan indikator-indikator lainnya yang berkaitan dengan aplikasi. Semua indikator penilaian aplikasi dalam penelitian ini disimbolkan dengan $V1, V2, \ldots, Vn$ untuk menjaga kerahasiaan data aplikasi perusahaan.

3.2.1 Variabel Prediktor

Dalam setiap penelitian terutama untuk analisis menggunakan metode Data Mining memerlukan adanya variabel-variabel yang berperan sebagai prediktor maupun target. Variabel prediktor (yang selanjutnya akan disebut fitur) X dari dataset aplikasi perusahaan terdiri dari 46 variabel. Dengan demikian, variabel prediktor/fitur yang telah disimbolkan dapat ditulis menjadi $V1, V2, V3, \ldots, V46$

3.2.2 Variabel Respon

Seperti yang telah dijelaskan sebelumnya bahwa penggunaan metode Data Mining membutuhkan variabel prediktor (fitur) dan juga variabel respon (target) (dalam *supervised learning*). Dalam penelitian ini akan dilakukan klasifikasi dan prediksi terhadap data yang sudah memiliki target menggunakan algoritma dari Data Mining, dimana algoritma tersebut dilatih (di *train*) terlebih dahulu agar dapat melakukan tugasnya (klasifikasi maupun prediksi). Berikut ini merupakan variabel respon Y (yang selanjutnya akan disebut label kelas) yaitu:

Tabel 3.1 Variabel Respon

Y	K_1	K_2	K_3	K_4
Label Kelas	M1	M2	M3	M4

dimana K adalah kelas yang dalam dataset terdiri dari empat kelas.

3.3 Tahapan Analisis

Langkah-langkah penelitian secara umum ditampilkan dalam bentuk diagram alir (*flowchart*) yang dapat dilihat pada Gambar 3.1. Adapun langkah-langkah analisis data dilakukan dengan cara sbb:

3.3.1 Pengumpulan Data

Pada tahapan ini, dilakukan pengumpulan dokumen-dokumen terkait cara kerja dan hubungan antar aplikasi. Data dikumpulkan dari berbagai sumber. Tahapan ini berguna untuk proses penentuan *primery key* dan *foreign key* pada masing-masing data atau tabel, sehingga proses transformasi dan penggabungan data dapat dilakukan dengan benar.

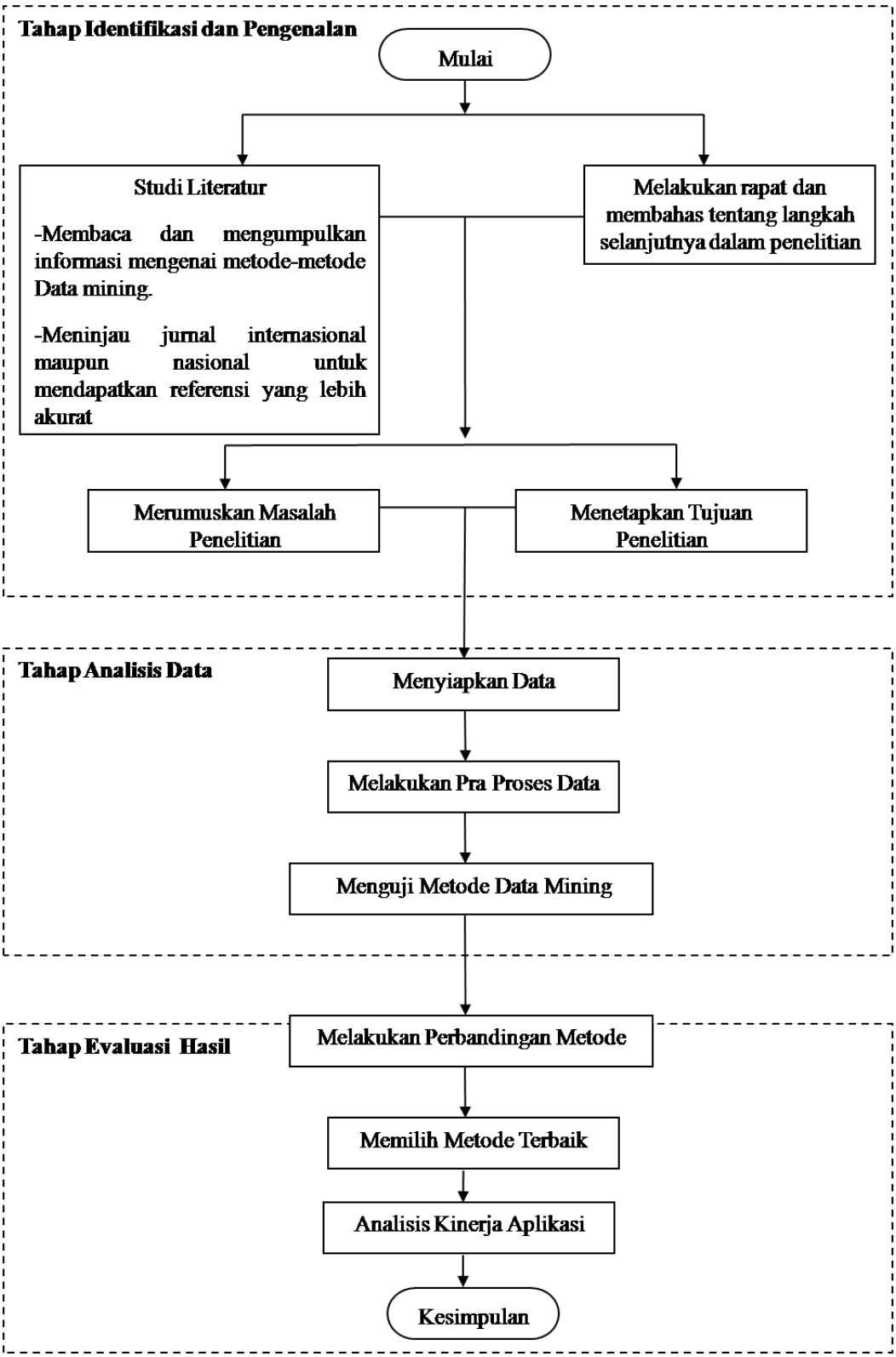

Fig. 3.1 Diagram Alir Penelitian

3.3.2 Persiapan Data

Persiapan data dilakukan untuk mengecek kelayakan data sebelum dilakukan proses selanjutnya. Pada penelitian ini, digunakan data mentah (*original data*) hasil dari proses ETL(*Extract Transform Load*) sehingga perlu adanya penyeleksian data terlebih dahulu sebelum melakukan pra proses data. Oleh karena itu, data dipilah kemudian dipilih untuk dibentuk menjadi dataset yang siap untuk diproses lebih lanjut.

Langah kedua yaitu preprocessing data yang dimulai dengan proses *cleaning data*. Terdapat banyak data yang hilang atau (*missing value*) dalam data dan karena data terdiri menjadi dua tipe berbeda yaitu tipe data numerik dan tipe data nominal, maka perlakuan yang diberikan untuk menangani masalah tersebut pun berbeda. Selain itu, terdapat nilai-nilai ekstrim dalam data yang lebih jelas akan dijelaskan pada bab selanjutnya.

3.3.3 Data Transformation

Pada tahapan ini dilakukan proses perubahan data yang dibutuhkan. Tranformasi data menggunakan teknik normalisasi skala data (hanya pada metode Data Mining tertentu) dengan tujuan data dapat diolah dengan baik.

3.3.4 Analisis Data

Setelah melakukan pra proses data, maka telah didapatkan data yang benar dan akurat sehingga penerapan metode-metode Data Mining dapat dijalankan dengan baik. Langkah-langkah analisis data untuk mencapai tujuan dari penelitian ini dilakukan dengan cara sbb:

1. Menerapkan metode Naive Bayes, Decision Tree, Random Forest, dan k-Nearest Neighbour,
2. Melihat nilai akurasi, presisi dan recall dari masing-masing metode.
3. Membandingkan metode-metode tersebut berdasarkan nilai akurasi, presisi dan recall serta waktu yang diperlukan untuk menjalankan metode.
4. Memilih metode terbaik berdasarkan nilai akurasi, presisi dan recall tertinggi dengan waktu yang cepat.
5. Analisis kinerja aplikasi dari hasil prediksi dari metode yang terbaik.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Persiapan Data

Seperti pada umumnya dalam sebuah penelitian, tahapan persiapan data ini dilakukan agar data telah benar-benar siap untuk dianalisis dengan metode-metode yang telah diusulkan sehingga dapat memberikan hasil yang lebih baik dan mengurangi kesalahan-kesalahan dalam pengolahan data. Adapun tahapan persiapan data ini termasuk tahap *preprocessing* dalam data mining yang telah dijelaskan di bagian sebelumnya. Tahap-tahap dalam persiapan data yaitu:

4.1.1 Pengumpulan dan Pendeskripsian Data

Mengingat sumber data berasal dari server dan aplikasi yang berbeda-beda maka untuk mengumpulkan data yang dibutuhkan, digunakan *tools* yang sudah ada yaitu SQL Server, SAP BO. Aplikasi tersebut digunakan untuk mengumpulkan, mentransformasi dan menggabungkan data. Dengan menggunakan aplikasi tersebut data-data yang tersebar dikumpulkan dan ditransformasi sesuai kebutuhan. Hasil dari proses tersebut berupa data set yang siap diolah pada tahapan selanjutnya.

Berdasarkan dataset yang didapatkan dari proses diatas, data tersebut terdiri dari 46 fitur dan 1 label dimana terdiri dari 4 kelas/group dan 2755 observasi. Data aplikasi yang digunakan perusahaan merupakan tipe data campuran (*mixed data type*) dimana terdapat fitur nominal, fitur biner maupun fitur numerik. Fitur-fitur tersebut terdiri dari:

1. 12 fitur nominal dengan 3 fitur biner. Fitur-fitur ini mendeskripsikan saran terhadap aplikasi, nama department yang mendukung aplikasi tersebut, tingkat layanan aplikasi dan juga status serta fungsinya.

2. 35 fitur lainnya berbentuk numerik. Fitur-fitur ini memuat nilai-nilai kritis dari aplikasi, jumlah insiden teknis maupun di luar teknis serta kapasitas aplikasi.

Sedangkan pada label terdiri dari 4 kelas yaitu C = [M1, M2, M3 dan M4] dimana masing-masing kelas dideskripsikan seperti pada Gambar 4.2:

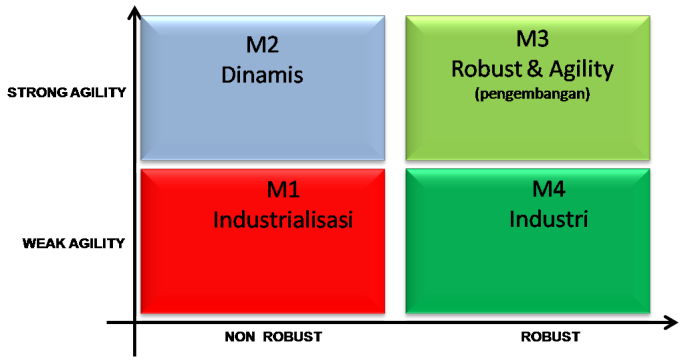

Fig. 4.1 Empat kelas label

Dari Gambar 4.2, suatu aplikasi dikategorikan ke dalam masing-masing kelas berdasarkan fitur-fitur penimbang yang dapat dijabarkan sbb:

1. Kelas M1 : aplikasi akan masuk ke dalam kelas M1 jika aplikasi tersebut *non robust* dan mempunyai *weak agility*. Kemudian aplikasi-aplikasi yang ada dalam kelas M1 harus dihapus karena tidak dapat memberikan manfaat bagi perusahaan atau hanya memberikan kerugian terhadap perusahaan. Aplikasi pada kelas ini disebut sebagai industrialisasi.

2. Kelas M2 : semua aplikasi pada kelas M2 disebut dinamis, artinya aplikasi tersebut masih dapat ditoleransi keberadaannya, sehingga tidak perlu dihapus dan dibiarkan berjalan apa adanya. Aplikasi akan dimasukkan ke dalam kelas ini jika aplikasi tersebut *non robust* dan *strong agility*.

3. Kelas M3 : aplikasi pada kelas ini merupakan aplikasi terbaik dengan kata lain bahwa aplikasi tersebut *robust* dan mempunyai *strong agility*. Aplikasi-aplikasi tersebut kemudian dikembangkan agar kinerjanya menjadi lebih baik lagi dan dapat memenuhi kebutuhan di masa yang akan datang.

4. Kelas M4 : pada kelas ini, aplikasi masih akan digunakan tetapi dengan catatan harus dilakukan perubahan dalam aplikasi tersebut. Misal terdapat aplikasi yang memakan waktu yang lama dalam pemrosesannya sehingga akan dicari penyebab dari lambatnya proses tersebut dan kemudian diperbaiki.

4.1.2 Praproses data

Setelah melakukan pengumpulan data, kemudian dilakukan tahap praproses data. Adapun langkah-langkahnya adalah sbb:

4.1.2.1 Pemilihan data

Pada tahap pengumpulan data, seringkali terjadi kesalahan dengan terambilnya data yang tidak relevan. Oleh karen itu, dilakukan penyeleksian terhadap data dengan hanya mengambil data yang relevan. Ada 1 fitur yang dihapus dikarenakan fitur tersebut berisi informasi yang tidak dibutuhkan dalam penelitian ini. Demikian pula terdapat aplikasi yang tidak tergolong ke dalam group/kelas tertentu, diakibatkan banyaknya informasi yang hilang. Oleh karena itu, aplikasi yang tidak memiliki kelas dihapus dari data. Dengan demikian, setelah melakukan penyeleksian, maka data keseluruhan terdiri dari $n+1$ atribut yaitu 45 atribut fitur, 1 atribut label terdiri dari 4 kelas/group dan 2620 observasi yang semuanya tercantum dalam Tabel 4.1. Data yang dipilih akan digunakan pada langkah berikutnya.

Tabel 4.1 Dataset Aplikasi Perusahaan

Y	Fitur Nominal	Fitur Biner	Fitur Numerik
Target Kelas (CADRAN) C = [M1, M2, M3 dan M4]	V1,V3,V5,V6,V7,V9,V10 dan V11	V2,V4 dan V8	V12-V44

Dari deskripsi data yang telah dijelaskan, dataset ini berbentuk multi kelas dapat dilihat pada Gambar 4.2. Dalam masalah klasifikasi multi kelas ini, terlihat bahwa jumlah aplikasi pada masing-masing kelas berbeda-beda. Presentasi terkecil berada pada kelas M3 mengartikan bahwa aplikasi yang *robust* dan *strong agility* sangat sedikit.

Dari Diagram tersebut jumlah aplikasi pada masing-masing kelas ditunjukkan pada Tabel 4.2 yaitu

Tabel 4.2 Jumlah Aplikasi pada Masing-Masing Kelas

M1	M2	M3	M4
644	349	93	1534

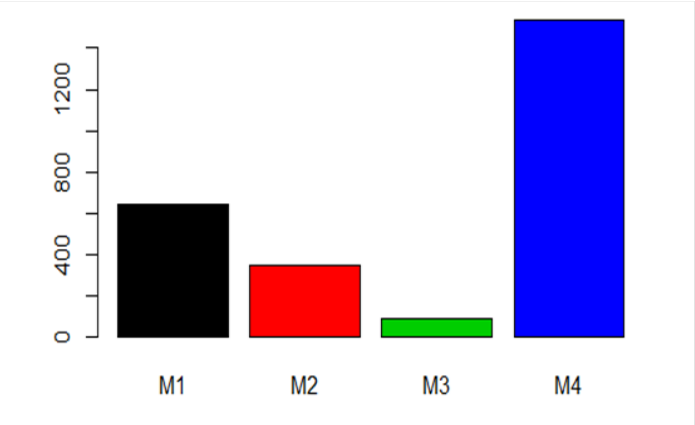

Fig. 4.2 Distribusi Frekuensi Label Kelas

4.1.2.2 Pembersihan Data

Tidak dapat dipungkiri bahwa bekerja dengan data mentah (asli) sangat berbeda dan lebih sulit dari apa yang diperkirakan. Banyak masalah data tidak dapat diselesaikan dengan mudah dan memerlukan perlakuan khusus, seperti dalam menangani adanya informasi-informasi yang hilang, adanya nilai-nilai yang berbeda skala pengukurannya atau ketidakkonsistennya data. Dengan demikian, pengolahan data dimulai dengan menyelidiki apakah ada kelainan pada variabel fitur dalam dataset. Untuk itu, pertama-tama dilakukan pendeteksian *missing value* dalam dataset dan hasilnya dapat dilihat pada Gambar 4.3.

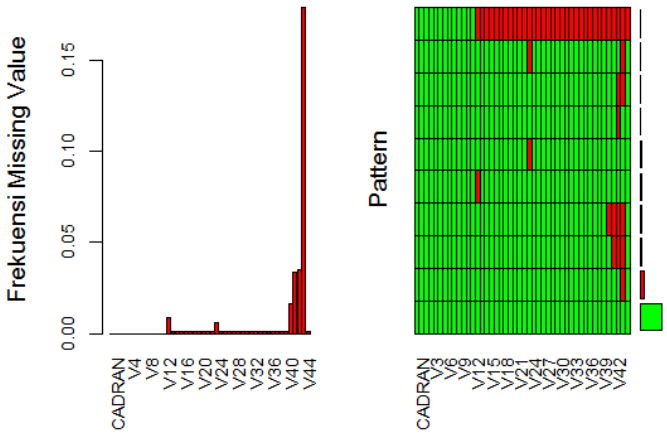

Fig. 4.3 Pola *Missing Value* Dataset

Gambar 4.3 menunjukkan bahwa adanya informasi/nilai yang hilang (*missing value*) pada dataset. Warna merah pada histogram mengindikasikan banyaknya nilai yang hilang tersebut. Hal ini tentu saja dapat mempengaruhi hasil prediksi dan keakuratan dalam melakukan

klasifikasi terutama jika menggunakan algoritma data mining yang tidak mentolerir adanya nilai yang hilang. Oleh sebab itu, dengan memperhatikan jenis fitur dataset yang berbeda-beda digunakan cara yang sudah tidak asing lagi untuk menangani masalah ini, yaitu:

1. **Modus** : untuk menangani *missing value* pada fitur nominal. Modus adalah nilai/objek yang paling banyak muncul. Pengimputan *missing value* dengan modus sangat tepat terutama jika jumlah *missing value*nya tidak terlalu banyak.

2. **Mean** : ini adalah cara yang paling populer dalam menangani nilai yang hilang untuk fitur bertipe numerik dan tentu saja dengan memperhatikan skala data pada masing-masing fitur.

Akan tetapi, sebelum menangani nilai yang hilang tersebut, diperiksa terlebih dahulu jika terdapat ketidakkonsistenan skala data. Dari hasil pemeriksaan, terdapat adanya tanda "?", "," dst yang bukan merupakan objek maupaun skala dari dataset perusahaan. Ketidakkonsistenan ini merupakan salah satu penyebab dataset terbaca *missing* serta gagalnya pengimputan *missing value* baik menggunakan modus maupun mean (rata-rata). Oleh karena itu, tanda-tanda yang tidak konsisten seperti itu dikosongkan sehingga memudahkan dalam pengimputan nilai yang hilang. Setelah melakukan "impute" pada dataset maka diperoleh hasil seperti pada Gambar 4.4

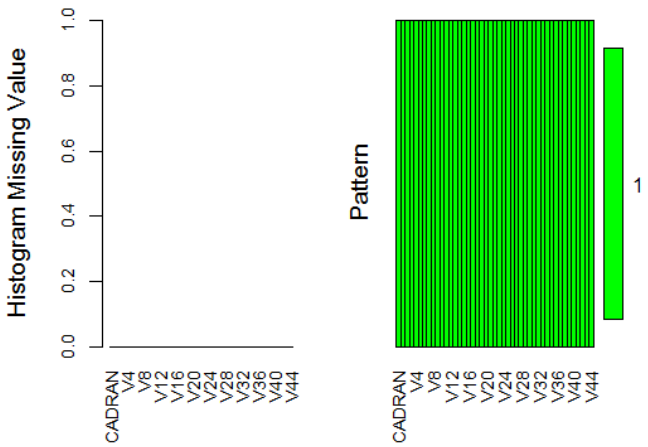

Fig. 4.4 Pola Dataset tanpa *Missing Value*

Selain masalah *missing value*, terdapat juga nilai ekstrim dalam dataset sehingga dilakukan analisis terhadap nilai ekstrim ini. Nilai ekstrim adalah objek data pada dataset yang tidak sesuai dengan perilaku umum objek data lainnya, sebagai contoh skala data pada variabel ke-44 (V44) berkisar antara 10-100 tetapi ada objek yang memiliki nilai 800-an.

Selain dari nilai yang tinggi, dikatakan data memiliki nilai ekstrim juga ketika suatu objek memiliki nilai yang terlalu kecil dimana nilai tersebut tidak sama dengan nilai objek-objek lainnya. Grafik nilai ekstrim ini disajikan pada Gambar 4.5

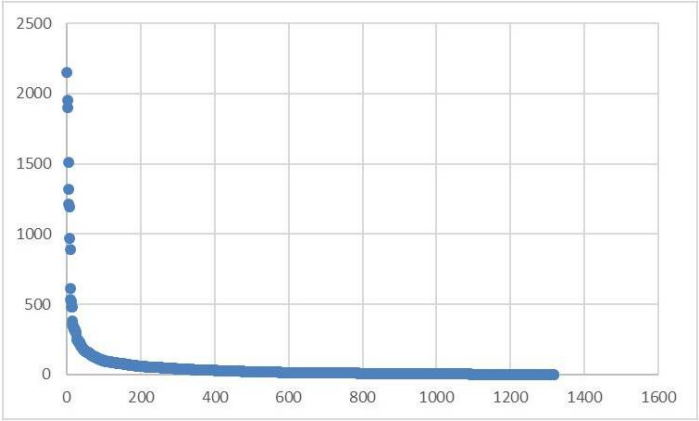

Fig. 4.5 Kurva Nilai Ekstrim

Ketika hasil identifikasi menunjukkan adanya data yang merupakan nilai ekstrim, maka yang bisa dilakukan adalah menghapus atau menghilangkannya pengamatan. Memang, jika tidak dihapus, akan berdampak untuk hasil terakhir.

4.1.3 Cross Validation

Pembagian data untuk pelatihan (*training*) dan evaluasi (*testing*) dilakukan dengan metode *cross validation* dimana proses diulangi beberapa kali dan dicari rata-rata hasil dari proses tersebut. Metode cross validation yang dipilih adalah K-fold cross validation yaitu membagi data menjadi K sampel yang berukuran sama dimana K-1 data digunakan untuk pelatihan (*training*) dan 1 lainnya untuk evaluasi serta iterasinya diulang sebanyak K kali. Nilai K yang dipilih pada penelitian ini adalah 10 (10-cross validation) karena 10 *folds* telah terbukti merupakan fold yang paling tepat untuk mendapatkan perkiraan kesalahan terbaik [43]. Untuk lebih jelasnya penggunaan metode 10-cross validation ini dapat dilihat pada Tabel 4.3:

Dari Tabel 4.3 dapat dengan jelas terlihat bahwa data dibagi menjadi 10 bagian yang sama dan setiap kali iterasi dilakukan, 1 bagian akan diperlakukan sebagai testing atau untuk evaluasi dan lainnya akan diproses untuk pelatihan.

Keuntungan dari K-cross validation adalah mengurangi bias karena teknik ini hanya menggunakan data parsial, sehingga dapat mengurangi kemungkinan overfitting ke dataset yang ada [27].

Tabel 4.3 Metode K-Cross Validation

Iterasi	Training dan Evaluasi									
	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
Iterasi 1	Test	Train	Train	Train	Train	Train	Train	Train	Train	Train
Iterasi 2	Train	Test	Train	Train	Train	Train	Train	Train	Train	Train
Iterasi 3	Train	Train	Test	Train	Train	Train	Train	Train	Train	Train
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
Iterasi 10	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	Test

4.1.4 Analisis Data

Pada analisis ini digunakan algoritma data mining untuk mengklasifikasikan dan memprediksi aplikasi perusahaan berdasarkan fitur-fitur yang telah ada.

Setelah langkah praproses data yang dilakukan sebelumnya, maka data telah benar-benar siap untuk dianalisis menggunakan algoritma-algoritma Data M ining. Karena dataset memiliki target dan target label kelas bertipe nominal maka akan dilakukan klasifikasi pada dataset dengan mengusulkan algoritma-algoritma pembelajaran dari *supervised machine learning algorithms*. Tujuan dari model klasifikasi ini adalah untuk mendapatkan beberapa hasil prediksi dan mengetahui tingkat keakuratan dari hasil tersebut. Ada berbagai macam metode yang dikembangkan untuk pembelajaran prediktif dari data. Ada situasi dimana metode tersebut sangat cocok pada dataset dan ada situasi dimana metode yang sama itu berkinerja sangat buruk. Oleh karena itu, harus berhati-hati dalam memilih metode klasifikasi yang sesuai dengan data dan tujuan. Dengan demikian, dipilih 3 metode pembelajaran dalam klasifikasi untuk pengujian dataset yaitu Bayesian Naive, Decision Tree, Random Forest dan k-Nearest Neighbour.

4.1.5 Naive Bayes

Perlu diingat kembali bahwa tujuan dari masalah klasifikasi adalah untuk menentukan keanggotaan dari kelas objek baru x_{new} (objek dalam kasus ini aplikasi), kemudian melakukan prediksi.

Seperti yang telah dijelaskan pada bab 2 bahwa metode Naive Bayes adalah metode klasifikasi yang didasarkan pada Teorema Bayes dengan asumsi independensi di antara prediktor (fitur). Dalam istilah sederhananya, klasifikasi Naive Bayes mengasumsikan

bahwa keberadaan fitur tertentu di kelas tidak terkait dengan keberadaan fitur lainnya karena masing-masing berkontribusi pada probabilitas [36]. Selain itu, Naive Bayes juga dikenal mengungguli metode klasifikasi lainnya bahkan sangat canggih. Tapi, karena dalam Naive Bayes tidak memilih fitur yang penting apapun maka dari metode ini tidak dapat dilihat fitur manakah yang paling informatif. Dalam pengklasifikasian aplikasi menggunakan metode ini, probabilitas dari masing-masing kelas dicari dengan menggunakan Persamaan 4.1 untuk variabel nominal. Namun, dataset dalam penelitian ini berbentuk campuran data (*mixed data*), maka untuk variabel numerik Persamaan yang digunakan adalah Persamaan 2.12 hingga Persamaan 2.18. Jika ada sampel baru yang ingin diketahui kelasnya maka dicari probabilitas dari sampel ini untuk masing-masing kelas sehingga kelas yang memiliki nilai probabilitas terbesar, itulah kelas dari sampel baru tersebut. Adapaun untuk variabel nominal persamaannya dapat dituliskan sbb:

$$P(Kelas|V_1,V_2,\ldots,V_{44}) = \frac{P(V_1,V_2,\ldots,V_{44}|Kelas).P(Kelas)}{P(V_1,V_2,\ldots,V_{44})} \tag{4.1}$$

Dengan menggunakan software WEKA dan 10-cross validation diperoleh hasil prediksi masing-masing kelas seperti tercantum pada Tabel 4.4.

Tabel 4.4 Confusion Matrix Metode Naive Bayes

Target	$M1$	$M2$	$M3$	$M4$
M1	185	138	26	0
M2	27	493	8	116
M3	9	10	49	25
M4	12	67	41	1414

Tabel 4.4 menunjukkan nilai prediksi dari masing-masing kelas baik yang diprediksi dengan benar maupun yang salah. Data aktual menunjukkan bahwa terdapat 93 aplikasi yang berada pada kelas M3 dan dari Tabel 4.4 diatas, terprediksi ada 93 juga aplikasi pada kelas M3. Namun dari 93 hasil prediksi ini, metode Naive Bayes hanya mampu memprediksi 49 aplikasi dengan benar dalam artian kelas aplikasi yang sama dengan data actualnya, sedangkan 44 lainnya adalah salah diprediksi yakni aplikasi tersebut bukanlah aplikasi yang berada pada kelas M3.

Setelah mendapatkan hasil prediksi untuk masing-masing kelas, kemudian akan dicari tingkat akurasi dari hasil prediksi metode klasifikasi Naive Bayes ini. Oleh karena itu, dengan menggunakan Persamaan 2.2, akurasi metode ini diperoleh. Dari Tabel 4.4 ini nilai akurasi prediksi dapat dihasilkan dengan menghitung jumlah aplikasi yang terprediksi dengan benar (aplikasi yang diprediksi sama dengan data aktualnya) pada masing-masing kelas dan dibagi

dengan banyaknya observasi. Dengan demikian cara menghitungnya (dalam persentase) adalah:

$$Akurasi = (\frac{185+493+49+1414}{2620})100\% = (0.817)100\% = 81.7\% \quad (4.2)$$

Nilai ini sama dengan nilai akurasi yang didapat dengan menggunakan WEKA. Namun, walaupun memiliki tingkat akurasi yang cukup baik yaitu 81.7%, nilai akurasi ini belum cukup untuk menyatakan bahwa metode Naive Bayes memprediksi dengan baik. Sebab, nilai akurasi tersebut merupakan akurasi prediksi secara keseluruhan. Sementara sangat penting juga untuk mengetahui ukuran keberhasilan prediksi melalui *precision* dan *recall* untuk masing-masing kelas yang hasilnya dapat dilihat pada Tabel 4.5.

Tabel 4.5 Precision and Recall Naive Bayes

Target	Precision	Recall
M1	0.794	0.530
M2	0.696	0.766
M3	0.395	0.527
M4	0.909	0.922

Dalam metode klasifikasi, *precision* adalah ukuran relevansi hasil, kemampuan model klasifikasi untuk mengidentifikasi hanya pada titik data yang relevan, sedangkan *recall* adalah ukuran dari berapa banyak hasil yang benar-benar relevan dikembalikan. Nilai *recall* yang semakin besar tidak dapat menunjukan suatu sistem baik atau tidak. Berbeda halnya dengan *precision* Semakin besar nilai *precision* yang dihasilkan, maka prediksi tersebut dapat dikatakan baik [30].

Cara menghitung *precision* pada kasus ini yaitu dengan menggunakan Persamaan 2.4. Dari persamaan tersebut diperoleh nilai *precision* untuk masing-masing kelas. Misal untuk memprediksi kelas M4 dengan merujuk pada Tabel 4.4 maka diperoleh:

$$Precision_{M4} = \frac{TP_{M4}}{TP_{M4} + FP_{M4}} \quad (4.3)$$

dimana:

- TP : Jumlah aplikasi/software pada kelas M4 yang diprediksi benar
- FP : Jumlah aplikasi/softwarre kelas M4 yang diprediksi tidak berada di kelas M4

$$Precision = \frac{1414}{1414 + (0 + 116 + 25)} \text{x} 100\% = 90\%$$

Precision dapat juga disederhanakan menjadi pembagian antara jumlah aplikasi yang diprediksi benar pada kelas M4 dengan jumlah semua aplikasi yang ditemukan, sedangkan *recall* untuk Naive Bayes diperoleh dari

$$Recall_{M4} = \frac{TP_{M4}}{TP_{M4} + FN_{M4}} \tag{4.4}$$

dimana:

- TP : Jumlah aplikasi/software pada kelas M4 yang diprediksi benar
- FN : Jumlah aplikasi/software yang bukan pada kelas M4 diprediksi berada di kelas M4.

$$Recall = \frac{1414}{1414 + (12 + 67 + 41)} \text{x} 100\% = 92\%$$

atau dapat juga disederhanakan menjadi pembagian antara jumlah aplikasi yang diprediksi benar pada kelas M4 dengan jumlah semua aplikasi M4 pada data aktual.

Dengan cara yang sama dihitung pula *precision* dan *recall* untuk kelas M1, M2 dan M3 yang hasilnya seperti pada Tabel 4.4. Dari hasil ini, rata-rata precision dan *recall* tertinggi yaitu pada kelas M4 dan rata-rata yang paling rendah ada pada kelas M3. Nilai rata-rata yang kecil ini dikarenakan jumlah yang tidak relevan lebih besar atau tidak seimbang dari jumlah yang relevan. Waktu untuk menjalankan algoritma Naive Bayes untuk klasifikasi dan prediksi aplikasi dengan menggunakan WEKA yaitu 0.44 detik.

4.1.6 Decision Tree

Pohon keputusan memecah kumpulan data menjadi himpunan bagian yang lebih kecil dan lebih kecil sementara pada saat yang sama sebuah pohon keputusan membentuk aturan-aturan keputusan. Hasil akhir adalah pohon dengan simpul keputusan dan simpul daun. Simpul pertama di pohon keputusan yang dihasilkan memberi tahukan fitur mana yang memiliki kekuatan prediktif paling tinggi.

Salah satu hal yang harus diperhatikan dalam metode pohon keputusan ini adalah begitu sulitnya menggambar diagram keputusan terutama untuk data multivariat atau data yang berdimensi besar. Namun, dengan menggunakan software WEKA, diagram hasil pohon keputusan dapat diperoleh dengan mudah.

Untuk mendapatkan pohon keputusan, setelah mempersiapkan dataset, langkah pertama yaitu menentukan akar dari pohon. Kemudian dataset dibagi berdasarkan fitur-fitur yang sesuai untuk dijadikan simpul daun. Permulaan dari tahap ini yaitu dengan melakukan

penyeleksian terhadap fitur dengan menggunakan Persaaman 2.20 dan Persamaan 2.21 untuk memperoleh nilai gain dari masing-masing fitur. Fitur yang memiliki nilai gain tertinggi akan menjadi *parent* bagi simpul-simpul selanjutnya. Kemudian mengulangi proses ini hingga semua tupel terpartisi dan berhenti ketika tidak ada fitur di dalam tupel yang dipartisi lagi atau dapat dikatakan semua cabang memiliki kelas yang sama. Hasil diagram pohon dari metode ini dapat dilihat pada Gambar 4.6.

```
J48 pruned tree
------------------

V17 <= 0
|   V18 <= 19.875
|   |   V14 <= 27
|   |   |   V16 <= 1
|   |   |   |   V15 <= 20: M4 (1519.0)
|   |   |   |   V15 > 20
|   |   |   |   |   V15 <= 29: M4 (15.0)
|   |   |   |   |   V15 > 29: M3 (10.0)
|   |   |   V16 > 1: M3 (34.0)
|   |   V14 > 27: M3 (49.0)
|   V18 > 19.875
|   |   V14 <= 29
|   |   |   V16 <= 1
|   |   |   |   V15 <= 28: M1 (157.0)
|   |   |   |   V15 > 28: M2 (5.0/1.0)
|   |   |   V16 > 1: M2 (9.0)
|   |   V14 > 29: M2 (24.0)
V17 > 0
|   V14 <= 29
|   |   V16 <= 1
|   |   |   V15 <= 30: M1 (486.0)
|   |   |   V15 > 30: M2 (53.0)
|   |   V16 > 1: M2 (78.0)
|   V14 > 29: M2 (181.0)

Number of Leaves  :     13
```

Fig. 4.6 Diagram Hasil Decision Tree

Dari Gambar 4.6, terlihat bahwa banyak simpul yang di *prune* untuk mendapatkan klasifikasi yang lebih baik. Simpul akar dari pohon ini adalah V17 yang artinya fitur V17 merupakan fitur yang paling informatif untuk membangun sebuah pohon.

Dengan menggunakan 10-*folds cross validation* maka hasil dari algoritma J48 dari metode pohon keputusan disajikan pada Tabel 4.6.

Tabel 4.6 Confusion Matrix Metode Decision Tree

Target	$M1$	$M2$	$M3$	$M4$
M1	349	0	0	0
M2	1	643	0	0
M3	0	0	93	0
M4	0	1	0	1533

Hasil klasifikasi dari metode pohon keputusan pada Tabel 4.6 menunjukkan bahwa prediksi menggunakan algoritma ini sangat baik untuk masing-masing kelas. Terlihat pada

tabel tersebut prediksi untuk kelas M3 adalah sempurna dimana dari 93 jumlah aplikasi yang ada dikelas M3 pada data aktual terprediksi juga ada 93 aplikasi dengan algoritma J48 dari pohon keputusan. Dari Tabel 4.6 dapat diperoleh nilai akurasi dari algoritma ini yaitu:

$$Akurasi = (\frac{349+643+93+1533}{2620})100\% = (0.9992)100\% = 99.92\%$$

Nilai akurasi 99.92% ini tentu lebih tinggi dari nilai akurasi metode sebelumnya yaitu memprediksi aplikasi pada masing-masing kelas menggunakan metode Naive Bayes. Nilai akurasi yang tinggi ini tentu selaras dengan hasil prediksi yang ditampilkan pada Confusion Matrix di Tabel 4.6 dimana pada kelas M1 hanya ada 1 FP yang mana merupakan FN pada kelas M2 dan 1 lainnya FP pada kelas M2 yang merupakan FN pada kelas M4, sedangkan untuk nilai *precision* dan *recall* dari algoritma ini yaitu disajikan pada Tabel 4.7:

Tabel 4.7 Precision and Recall Decision Tree

Target	Precision	Recall
M1	0.997	1.000
M2	0.998	0.998
M3	1.000	1.000
M4	1.000	0.999

Nilai *precision* dan *recall* dari algoritma ini wajar saja sangat baik dimana hasil prediksi yang jelas terlihat pada Tabel 4.6. Ketepatan prediksi yang menghasilkan akurasi, presisi dan recall yang tinggi tersebut dikarenakan algoritma J48 menangani data yang bertipe nominal dan numerik dengan sangat baik, terutama dataset dalam penelitian ini sudah tidak mengandung nilai yang hilang, overfitting yang terjadi karena adanya noise atau data yang tidak relevan, yang mana masalah-masalah ini telah ditangani pada tahap pra proses data. Hal ini mengakibatkan pemangkasan cabang (prune) di pohon mejadi lebih mudah karena pohon sudah tidak memiliki banyak cabang dan menjadi lebih seimbang dan tentu saja ia berdampak pada pembangunan pohon yang lebih cepat dan hasil dari klasifikasinyapun menjadi semakin akurat. Selain itu, algoritma J48 merupakan algoritma yang cepat diantara algoritma-algoritma lainnya [16] dalam pohon keputusan yang terbukti pada penelitian ini hanya membutuhkan 0.71 detik untuk mendapatkan hasil prediksi yang tepat.

4.1.7 Random Forest

Metode Random Forest mampu menangani masalah klasifikasi maupun regresi. Untuk memprediksi sample baru dalam metode ini, sampel baru tersebut dimasukkan ke dalam

pohon keputusan yang ada untuk ditentukan kelasnya. Prosedur ini dilakukan berulang kali terhadap keseluruhan pohon keputusan yang tergabung di dalam Random Forest. Kelas dari sample baru tersebut ditentukan dengan cara melihat *"voting"* hasil kelas mayoritas oleh seluruh pohon keputusan di dalam metode ini. Hasil prediksi dari Random Forest disajikan pada Tabel 4.8.

Tabel 4.8 Confusion Matrix Metode Random Forest

Target	$M1$	$M2$	$M3$	$M4$
M1	316	31	2	0
M2	10	622	0	12
M3	2	0	42	49
M4	0	3	0	1531

Dari hasil prediksi tersebut terlihat bahwa prediksi untuk masing-masing kelas terbilang cukup baik. Terdapat nol kesalahan prediksi yang terjadi pada baris kelas M4 yang menandakan FN kelas ini sedikit. Secara keseluruhan ada 109 aplikasi dari 2620 aplikasi yang diprediksi salah. Keakuratan penerapan metode Random Forest untuk mengklasifikasikan aplikasi ke dalam beberapa kelas dihitung sbb:

$$Akurasi = \left(\frac{316+622+42+1531}{2620}\right)100\% = (0.9583)100\% = 95.83\%$$

Pada metode Decision Tree diperoleh akurasi sebesar 99.92% dan pada metode ini diperoleh akurasi sebesar 95.83%. Mengingat bahwa Random Forest adalah kumpulan dari Decision Tree, maka diharapkan metode ini akan mendapatkan akurasi yang lebih baik. Namun, pada kenyataannya dalam penelitian ini, metode Random Forest jika dilihat dari segi tingkat keakuratan prediksi tidak jauh lebih baik dari Decision Tree.

Kemudian, sama halnya dengan metode-metode yang diusulkan sebelumnya maka untuk metode Random Forest nilai *precision* dan *recall*nya disajikan pada Tabel 4.9.

Tabel 4.9 Precision and Recall Random Forest

Target	Precision	Recall
M1	0.963	0.905
M2	0.948	0.966
M3	0.955	0.452
M4	0.962	0.998

Tabel 4.9 menunjukkan bahwa baik nilai precision maupun recall tidak jauh berbeda, kecuali pada kelas M3. Nilai recallnya sangat berbeda dengan nilai presisi, yang mengartikan bahwa baik FN dan FP dalam prediksi kelas ini tidak seimbang.

4.1.8 k-Nearest Neighbour

Pada algoritma k-tetangga terdekat, pemilihan nilai k diperoleh dari $k = \sqrt{n}$ yang mana bersesuaian dengan error terkecil yang mengacu pada hasil *cross validation*. Namun, hasil klasifikasi aplikasi untuk masing-masing kelas tersebut sangat buruk. Untuk itu, kemudian semua fitur pada dataset dinormalisasi terlebih dahulu kemudian dilakukan percobaan sekali lagi untuk mengklasifikasi aplikasi. Tingkat akurasi kali ini tidak jauh berbeda dengan sebelumnya, tetapi nilai presisi dan recall semakin baik. Setelah mengamati dan mencoba dengan nilai k berbeda, baik tingkat akurasi maupun presisi dan recall tidak mengalami perubahan yang cukup berarti. Oleh karena itu, dipilih $k = 4$ yang menghasilkan nilai yang lebih baik dari percobaan-percobaan sebelumnya. Hasil ini dapat dlihat pada Tabel 4.10.

Tabel 4.10 Confusion Matrix Metode k-Nearest Neighbour

Target	$M1$	$M2$	$M3$	$M4$
M1	173	99	11	66
M2	85	327	12	220
M3	16	17	11	49
M4	30	164	29	1311

Dari Tabel 4.10 tersebut didapatkan akurasi untuk algoritma ini adalah

$$Akurasi = \left(\frac{173+327+12+1311}{2620}\right)100\% = (0.6954)100\% = 69.54\%$$

Sejauh ini dapat dikatakan bahwa nilai akurasi dari metode k-tetangga terdekat adalah yang paling kecil dari metode lainnya yaitu hanya 69.54% dari data, sedangkan untuk nilai *precision* dan *recall*nya disajikan pada Tabel 4.11.

Tabel 4.11 Precision and Recall k-Nearest Neighbour

Target	Precision	Recall
M1	0.569	0.496
M2	0.539	0.508
M3	0.175	0.118
M4	0.796	0.855

Melihat Tabel 4.11, dapat dicatat bahwa hasil prediksi untuk kelas M3 memiliki nilai *precision* dan *recall* yang paling kecil atau sangat buruk. Hal ini normal ketika melihat hasil prediksi pada Tabel 4.10 yang mana algoritma kNN hanya mampu memprediksi dengan benar 11 aplikasi dari 93 aplikasi yang ada dikelas M3 pada data aktual. Namun, ada hal yang menarik dari penerapan algoritma ini dalam kasus yang ada yaitu algoritma tersebut hanya membutuhkan waktu 0.05 detik untuk mengeksekusi algirtma di software WEKA bahkan saat melakukan 10-cross validation sekalipun. Dengan demikian dapat disimpulkan bahwa algoritma tercepat dari ketiga algoritma adalah algoritma k-Nearest Neighbour.

4.1.9 Perbandingan Metode Data Mining

Hasil perbandingan ke-empat metode yang diusulkan untuk memprediksi kelas aplikasi dari dataset dalam penelitian ini menggunakan software WEKA disajikan pada Tabel 4.12.

Tabel 4.12 Perbandingan Akurasi Metode

Metode	Akurasi(%)	Waktu(detik)
Naive Bayes	90	0.44
Decision Tree	99.92	0.71
Random Forest	95.83	6.33
k-NN	69.54	0.05

Pada Tabel 4.12 menunjukkan bahwa nilai akurasi terbaik diperoleh dari metode Decision Tree yaitu 99.92%, sedangkan nilai akurasi yang tidak terlalu baik adalah dari metode k-Nearest Neighbour yang hanya 69.54. Namun, dalam menjalankan algoritma di *software* WEKA, algoritma k-Nearest Neighbour merupakan algoritma tercepat yang hanya membutuhkan 0.05 detik, sementara algoritma Random Forest merupakan yang paling lambat yang membutuhkan waktu 6.33 detik.

Dari hasil penelitian ini dapat disimpulkan bahwa metode terbaik untuk mengklasifikasikan dan memprediksi aplikasi yang digunakan perusahaan yaitu metode Decision Tree dengan nilai akurasi tertinggi dengan waktu yang cukup cepat.

4.1.10 Analisis Kinerja Aplikasi

Setelah melakukan klasifikasi dan prediksi dengan metode Data Mining dan didapatkan metode terbaik yaitu Decision Tree dimana analisis ini menggunakan data tahun 2014-2016. Dari hasil klasifikasi dan prediksi ini diperoleh semua nama-nama aplikasi yang masuk ke dalam masing-masing kelas/group atau zona yaitu M1, M2, M3 dan M4. Selanjutnya

dilakukan analisis prediksi untuk data tahun 2017. Analisis ini bertujuan untuk melihat peningkatan aplikasi dan perpindahan aplikasi dari zona yang satu ke zona lainnya.

Seperti yang telihat pada Gambar 4.7, sebagian besar aplikasi atau sekitar 74% aplikasi berada di zona M1 dan M2. Pada M1, terdapat 175 aplikasi yang tetap berada di zona tersebut, sedangkan 81 aplikasi lainnya keluar dari zona tersebut. Dari 264 aplikasi yang berada dalam zona ini sebelumnya, ada 8 aplikasi yang sudah tidak digunakan lagi oleh perusahaan. Dengan kata lain, aplikasi tersebut dihentikan penggunaannya atau fungsinya digantikan dengan aplikasi lainnya. Di sisi lain, ada 142 aplikasi yang masuk dalam zona M1, dari jumlah total 322 aplikasi yang ada di zona ini. Dari jumlah tersebut ada 5 aplikasi baru.

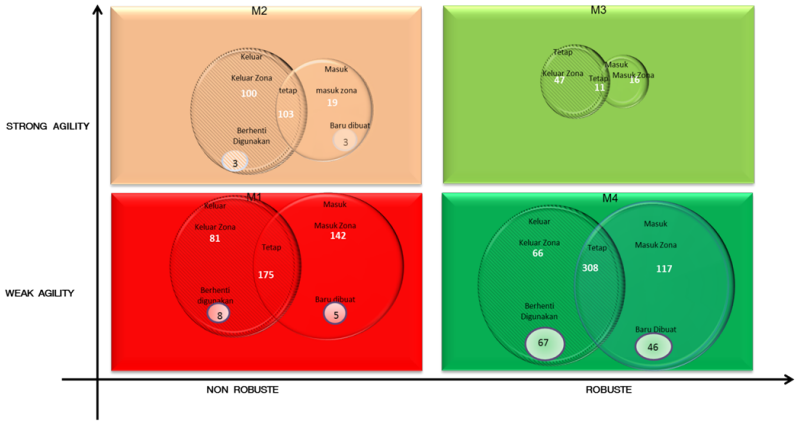

Fig. 4.7 Perbandingan Kondisi Aplikasi Tahun 2016 dan Hasil Prediksi Tahun 2017

Beralih ke zona M2, pada zona ini terdapat 103 aplikasi yang tetap atau tidak berubah dari tahun sebelumnya dari total ada 125 aplikasi. Dari jumlah tersebut terdapat 19 aplikasi yang masuk ke zona M2 dan terdapat 3 aplikasi yang baru dibuat. Berlawanan dengan itu, dari 206 aplikasi yang ada di zona ini sebelumnya, terdapat 100 aplikasi yang keluar dari zona M2 dan 3 dari jumlah total tersebut merupakan aplikasi yang sudah tidak digunakan.

Zona M3 merupakan zona yang paling kecil jika dilihat dari jumlah aplikasi yang ada didalamnya. Zona ini adalah zona paling bagus diantara yang lainnya, dimana aplikasi yang ada di dalam zona ini merupakan aplikasi yang mempunyai *strong agility* dan *robust*. Dalam zona M3 terdapat 11 aplikasi yang tidak berpindah ke zona lainnya, sedangkan ada 47 aplikasi yang keluar dari zona ini, yang menandakan aplikasi tersebut menurun performanya. Selain itu, ada 16 aplikasi yang masuk ke dalam zona ini, hal tersebut menandakan aplikasi tersebut meningkat dari segi performannya.

Zona yang terakhir yaitu M4, pada zona ini jumlah aplikasi lumayan banyak. Ada sekitar 308 aplikasi yang tetap pada zona ini dan 66 aplikasi lainnya keluar dari zona ini. Dari

jumlah aplikasi yang berada di zona M4 sebelumnya terdapat 67 aplikasi yang sudah tidak digunakan, sedangkan ada 46 aplikasi yang baru dibuat masuk dalam zona M4.

Kemudian beralih ke perpindahan aplikasi ke zona lain yang dapat dilihat pada Gambar 4.8. Dari Gambar 4.8, zona tujuan terbanyak adalah M4 yang menandakan 70 aplikasi tersebut mengalami perbaikan atau peningkatan dari segi performa dari periode sebelumnya.

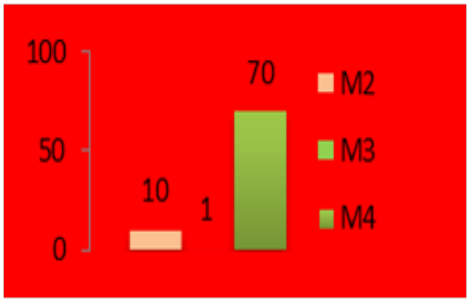

Fig. 4.8 Detail Aplikasi yang Keluar dari M1

Selain itu, terdapat 1 aplikasi yang mengalami peningkatan yang signifikan, yaitu berpindah dari zona M1 ke zona yang paling bagus yaitu M3. Terdapat 10 aplikasi yang juga mengalami perbaikan, yaitu berpindah dari zona M1 ke zona M2 yang menandakan aplikasi tersebut sudah termasuk aplikasi yang memiliki *strong agility*. Setelah itu, dilanjutkan dengan melihat aplikasi yang mengalami penurunan yang tersajikan pada Gambar 4.9.

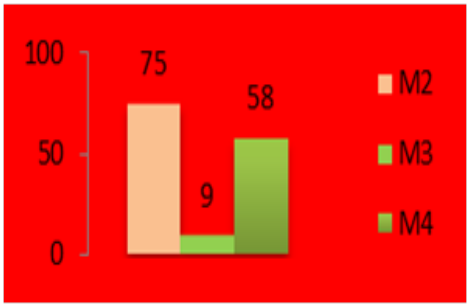

Fig. 4.9 Detail Aplikasi yang Masuk M1

Sekarang berpindah ke aplikasi yang mengalami penurunan performa. Dibandingkan jumlah aplikasi yang keluar dari zona M1, aplikasi yang masuk jauh lebih banyak yaitu sebanyak 142 aplikasi. Terdapat 9 aplikasi yang mengalami penurunan performa yang paling signifikan yaitu berpindah dari M3 ke M1, mengingat zona M1 merupakan zona yang paling tidak baik dibandingkan dengan zona lainnya. M4 juga merupakan penyumbang terbanyak kedua setelah M2 yaitu sebanyak 58 aplikasi, sedangkan pada M2 ada 75 aplikasi.

Seperti yang terlihat pada Gambar 4.10, zona yang paling banyak menjadi tujuan aplikasi adalah M1. Hal tersebut menandakan 75 aplikasi yang berpindah dari M2 ke M1 mengalami

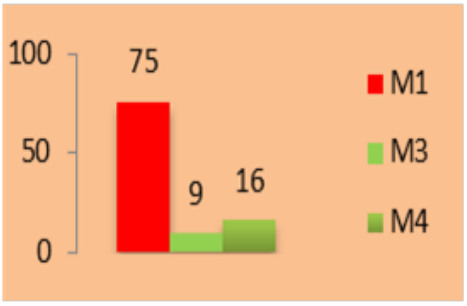

Fig. 4.10 Detail Aplikasi yang Keluar dari M2

penurunan *agility* dari sebelumnya termasuk aplikasi yang *strong agility*, sekarang memiliki *weak agility*. Selain itu, ada juga aplikasi yang mengalami perbaikan, yaitu 9 aplikasi berpindah ke M3 dan 16 aplikasi berpindah ke M4, sedangkan aplikasi yang masuk ke dalam zona M2 disajikan pada Gambar 4.11.

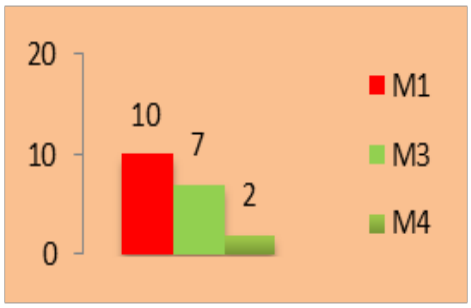

Fig. 4.11 Detail Aplikasi yang Masuk dari M2

Seperti halnya jumlah aplikasi yang keluar, jika mengacu pada Gambar 4.11, jumlah aplikasi yang masuk paling banyak berasal dari M1 walaupun dari segi jumlah tidak sebanyak aplikasi yang keluar yaitu hanya 10 aplikasi yang berpindah dari M1 ke M2, sedangkan aplikasi yang masuk sebanyak 75 aplikasi. Dengan demikian, hanya sekitar 13% aplikasi yang mengalami perbaikan dibandingkan aplikasi yang mengalami penurunan performa. Selain itu, ada 9 aplikasi yang mengalami penurunan performa yaitu 7 aplikasi yang berasal dari M3 dan 2 aplikasi yang berasal dari M4. Dilanjutkan pada zona M3 yang dapat dilihat pada Gambar 4.12

Seperti yang terlihat pada Gambar 4.12, ada 47 aplikasi yang mengalami penurunan performa walapun sebagian besar penurunan performa tidak terlalu signifikan karena aplikasi berpindah dari M3 ke M4 yaitu sebanyak 31 aplikasi, sedangkan 9 aplikasi lainnya mengalami penurunan performa yang signifikan yaitu dari M3 ke M1 dan ada 7 aplikasi berpindah ke M2. Untuk aplikasi yang masuk dapat dilihat pada Gambar 4.13.

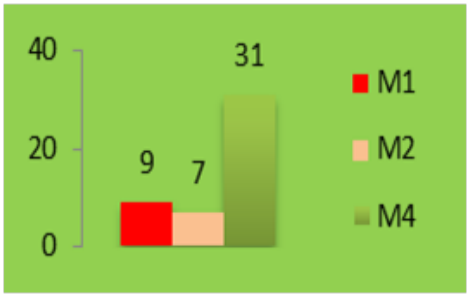

Fig. 4.12 Detail Aplikasi yang Keluar dari M3

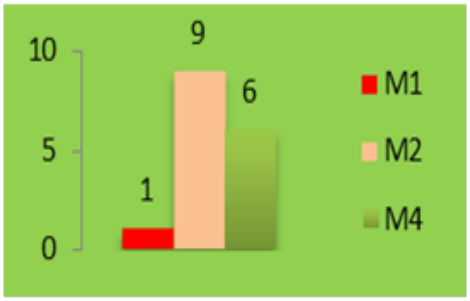

Fig. 4.13 Detail Aplikasi yang Masuk dari M3

Dapat dilihat pada Gambar 4.13, ada 16 aplikasi yang mengalami peningkatan performa, dari jumlah tersebut ada 9 aplikasi yang berasal dari M2, 6 aplikasi berasal dari M4 dan hanya 1 aplikasi yang berasal dari M1.

Setelah menganalisis perpindahan aplikasi, selanjutnya akan dibahas hasil analisis dari jumlah karyawan yang menangani aplikasi berdasarkan zona M1 sampai M4. Analisis ini bertujuan untuk mendapatkan referensi yang akurat mengenai jumlah karyawan yang dibutuhkan dalam menangani aplikasi perusahaan, sehingga dapat menghindari adanya kelebihan ataupun kekurangan jumlah karyawan. Hal ini mengingat untuk menangani setiap aplikasi membutuhkan jumlah karyawan yang berbeda-beda. Hasil analisis dapat dilihat pada Gambar 4.14 yaitu jumlah karyawan yang menangani aplikasi berdasarkan zona aplikasi tersebut.

Berdasarkan Gambar 4.14, sekitar 74% dari total ada 963 karyawan yang bertangung jawab atau menangani aplikasi yang berada di zona M1 dan M2. Dari 322 aplikasi yang berada di zona M1 ada 278 karyawan yang bertangung jawab pada aplikasi tersebut dan 74 atau sekitar 27% menangani permasalahan AP serta 177 atau 64% dari 278 menangani permasalahan AR. Sisanya menangani masalah yang lain misalnya pemeliharaan seperti reboot server, memastikan suplay energi listrik ke server dan lain sebagainya, sedangkan pada zona M2 membutuhkan banyak sumber daya manusia, ada 435 karyawan yang dibutuhkan untuk 125 aplikasi. Jumlah ini terbanyak jika dibandingkan dengan zona yang lainnya. Ada

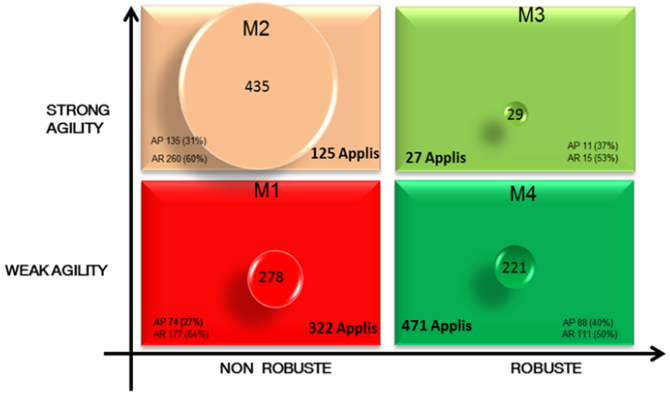

Fig. 4.14 Sebaran Karyawan Berdasarkan Aplikasi

sekitar 4 karyawan yang dibutuhkan untuk setiap aplikasinya. Dari 435 karyawan, ada 135 karyawan yang menangi AP dan 260 yang menangani permasalah AR.

Selanjutnya pada zona M3 terdapat 27 aplikasi yang membutuhkan 29 karyawan untuk memastikan aplikasi tersebut berjalan sebagaimana mestinya. Hampir hanya membutuhkan 1 orang saja pada setiap aplikasi. Dari 29 karyawan, 37% atau 11 karyawan yang menangani permasalahan AP dan 15 karyawan yang menangani permasalah AR. Diharapkan pada kemudian hari, semua aplikasi dapat berada di dalam zona ini, karena selain baik untuk performa perusahaan juga dapat menghemat biaya pemeliharaan aplikasi mengingat jika aplikasi berada dalam zona ini pasti aplikasi tersebut robust dan juga memiliki *strong agility*.

Beralih ke M4, pada zona ini aplikasi memiliki *weak agility* artinya perubahan dalam aplikasi tersebut sangat kecil sehingga membutuhkan semakin sedikit sumber daya manusia. Untuk 471 aplikasi hanya dibutuhkan 221 karyawan. Memiliki weak agility sangat baik untuk memangkas biaya pemeliharaan. Namun, tidak baik untuk manfaat dari aplikasi tersebut karena aplikasi tersebut tidak mampu beradaptasi dengan kebutuhan perusahaan secara baik artinya aplikasi tersebut kurang fleksibel. Dari 221 karyawan, 88 karyawan menangani AP dan 111 menangani AR.

Selanjutnya akan dianalisis perbandingan jumlah karyawan dari tahun sebelumnya berdasarkan aktivitas yang dilakukan. Terdapat 4 kelompok aktivitas yang akan dibandingkan yaitu AP, INC, MCO dan support. Lebih jelasnya dapat dilihat pada Gambar 4.15.

Seperti yang terlihat pada Gambar 4.15, pada Zona M1 peningkatan jumlah karyawan paling tinggi ada pada kelompok AP yaitu sebanyak 48 karyawan dari sebelumnya 43 karyawan sekarang menjadi 91 karyawan yang menangangi permasalahan AP, sedangkan peningkatan paling sedikit pada kelompok MCO yaitu sebanyak 16 karyawan dari 36 karyawan sekarang menjadi 52 karyawan yang menangani permasalahan MCO atau pemeliharaan aplikasi yang sedang berjalan.

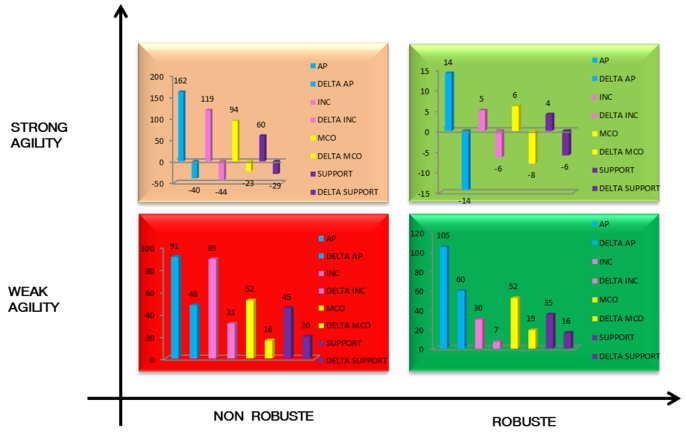

Fig. 4.15 Detail aktivitas Karyawan

Berbeda dengan Zona M1 yang semua aktivitas mengalami kenaikan, pada zona M2 semua aktivitas mengalami penurunan jumlah karyawan. Penurunan yang paling signifikan terdapat pada kelompok INC atau insiden pada aplikasi. Pada aktivitas ini jumlah karyawan mengalami penurunan sebanyak 44 karyawan dari sebelumnya dibutuhkan 163 karyawan sekarang hanya dibutuhkan 119 karyawan. Selain itu, kelompok aktivitas AP juga mengalami penurunan yang tidak kalah dengan INC yaitu sebanyak 40 karyawan. Total ada penurunan 139 karyawan pada zona M2.

Selaras dengan M2, pada M3 juga terdapat penurunan jumlah karyawan dari yang sebelumnya 63 karyawan menjadi 29 karyawan artinya terdapat penurunan 34 karyawan pada zona ini. Kelompok aktivitas AP menjadi menyumpang terbesar penurunan karyawan sebanyak 14 karyawan dari sebelumnya 28 karyawan menjadi 14 karyawan, sedangkan pada MCO terdapat penurunan 8 karyawan dari sebelumnya 14 menjadi hanya 6 karyawan. Seperti halnya M1, M4 juga mengalami peningkatan karyawan di semua aspek walaupun jumlah peningkatannya tidak sebanyak M1. AP masih penyumbang terbanyak peningkatan tersebut, dari total terdapat 102 peningkatan karyawan terdapat 60 dari total tersebut berasal dari AP, sedangkan MCO terdapat sedikit peningkatanya itu sebanyak 7 karyawan.

BAB 5

KESIMPULAN DAN SARAN

Pada bab ini diberikan kesimpulan dari hasil dan pembahasan yang diperoleh dari hasil penelitian. Selain itu, dalam bab ini juga diberikan saran untuk pengembangan yang lebih lanjut dari penelitian yang telah dilakukan.

5.1 Kesimpulan

Pada penelitian ini, ditunjukkan bahwa metode Data Mining berupa Naive Bayes, Decision Tree, Random Forest dan k-Nearest Neighbour dapat mengklasifikasikan dan memprediksi aplikasi yang digunakan perusahaan dengan hasil yang akurat. Gagasan pengusulan metode Data Mining ini adalah dikarenakan banyak penelitian yang telah membuktikan manfaat dari metode tersebut, tetapi karena masing-masing kasus dalam data berbeda-beda tidak banyak juga yang gagal menerapkannya. Dari hasil penelitian, metode Decision Tree merupakan metode terbaik dengan tingkat akurasi tertinggi yaitu 99.92% dan kecepatan waktu yang digunakan dalam menjalankan aplikasi yang hanya 0.71. Dari hasil prediksi dengan nilai akurasi yang baik ini dapat pula disimpulkan bahwa pengumpulan data dengan teknik Business Intelligence (BI) menghasilkan data yang relevan dan lebih mudah untuk dianalisis.

Kemudian setelah melakukan pengklasifikasian dan menggunakan hasil yang diperoleh dari Decision Tree yang berisi nama-nama aplikasi pada masing-masing kelas/zona, analisis dilanjutkan dengan mengidentifikasi perpindahan aplikasi dari zona satu ke zona lainnya. Kemudian terakhir melakukan analisis terhadap jumlah karyawan yang menangani setiap aplikasi.

5.2 Saran

Penelitian ini berfokus pada hasil perbandingan metode berdasarkan nilai akurasi, *precision* dan juga *recall* dari keempat metode Data Mining. Untuk penelitian selanjutnya diharapkan dapat mengevaluasi hasil berdasarkan nilai F-measure dan menampilkan kurva ROC (*Receiver operating characteristic*) dari masing-masing metode. Selain itu, karena data bertipe multi kelas diharapkan di penelitian selanjutnya dapat menerapkan metode *Multiclass Classifier* dan metode Data Mining lainnya untuk mendapatkan perbandingan yang lebih baik.

Daftar Pustaka

[1] Abdel-Aal, H. K., Bakr, B. A., and Al-Sahlawi, M. A. (1992). *Petroleum Economics and Engineering, Ed.2*. Marcel Dekker, Inc, New York.

[2] Bayrak, T. (2015). A review of business analytics: A business enabler or another passing fad. *Procedia-Social and Behavioral Sciences*, 195.

[3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[4] Bustami (2014). Penerapan algoritma naive bayes untuk mengklasifikasi data nasabah asuransi. *Jurnal Informatika*, 8(1).

[5] Chandra, B., Gupta, M., and Gupta, M. P. (2007). Robust approach for estimating probabilities in naive-bayes classifier. In *Pattern Recognition and Machine Intelligence*, pages 11–16, Berlin, Heidelberg. Springer Berlin Heidelberg.

[6] Coppin, B. (2004). *Artificial Intelligent Illuminated*. Jones and Bartlett, London, UK.

[7] Cortez, P. and Santos, M. F. (2013). Knowledge discovery and business intelligence. *Expert Systems*, 30.

[8] Darudiato, S., Santoso, S. W., and Wiguna, S. (2010). Business intelligence: Konsep dan metode. *CommIT*, 4(1).

[9] Dzikrulloh, N. N., Indriati, and Setiawan, B. D. (2017). Classification algorithm in data mining: An overview. volume 1.

[10] Gibson, J. M., Ivancevich, J. L., and Donelly, J. H. (1994). *Organizations: Behavior, Structure, Processes (8th ed.)*. Richard D. Irwin, Burr Ridge, IL.

[11] Hahn, R. W. (2004). *Government Policy towards Open Source Software*. Tata McGraw-Hill, New Delhi.

[12] Han, J., Camber, M., and Pei, J. (2012). *Data Mining: Concepts and Techniques Ed. 3*. Morgan Kaufmann, USA.

[13] Hsu, C.-C., Huang, Y.-P., and Chang, K.-W. (2008). Extended naive bayes classier for mixed data. *Expert Systems with Applications*, 35:1080–1083.

[14] Imelda (2013). Bussines intelligence. *Majalah Ilmiah UNIKOM*, 11(1).

[15] Jawadekar, W. S. (2002). *Software Engineering Principles and Practice*. Brookings Institution Press, Washington.

[16] Jayanti, N., Puspitodjati, S., and Elida, T. (2018). Teknik klasifikasi pohon keputusan untuk memprediksi kebangkrutan bank berdasarkan rasio keuangan bank.

[17] Kalsum, U. (2009). Penggunaan pohon keputusan (decision tree) untuk pengambilan keputusan dalam penerimaan pegawai. Master's thesis, Universitas Negeri Islam Sultan Syarif Kasim Riau.

[18] Karim, M. and Rahman, R. M. (2013). Decision tree and naïve bayes algorithm for classification and generation of actionable knowledge for direct marketing. *Journal of Software Engineering and Applications*, 6.

[19] Kashyap, P. (2017). *Machine Learning For Decision Makers*. apress, Bangalore, India.

[20] Kataria, A. and Singh, M. D. (2013). A review of data classification using k-nearest neighbour. *International Journal of Emerging Technology and Advanced Engineering*, 3.

[21] Kesavaraj, G. and Sukumaran, S. (2013). A study on classification techniques in data mining. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–7.

[22] Khamis, H. S., Cheruiyot, K. W., and Kimani, S. (2014). Application of k- nearest neighbour classification in medical data mining. *International Journal of Information and Communication Technology Research*, 4(4).

[23] Kunang, Y. N. and Andri, A. (2013). Implementasi teknik data mining untuk memprediksi tingkat kelulusan mahasiswa. *Seminar Nasional Informatika*.

[24] Kusrini and Luthfi, E. T. (2009). *Algoritma Data Mining*. ANDI, Yogyakarta.

[25] Larose, D. T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley Sons, Inc., Hoboken, New Jersey.

[26] Larose, D. T. and Larose, C. D. (2015). *Data Mining and Predictive Analytics*. John Wiley Sons, Inc., Hoboken, New Jersey.

[27] Natingga, D. (2017). *Data Science Algorithm in a Week*. Packt Publishing, Birmingham.

[28] Neelamegam, S. and Ramara, E. (2013). Classification algorithm in data mining: An overview. volume 4 issue 8.

[29] Nofriansyah, D. (2014). *Konsep Data Mining Vs Sistem Pendukung Keputusan Ed.1*. Deepublish, Yogyakarta.

[30] Novianti, K. D. P., Setiawan, N. A., and Kusumawardani, S. S. (2015). Peningkatan nilai recall dan precision pada penelusuran informasi pustaka berbasis semantik. *Konferensi Nasional Sistem Informatika*.

[31] Nurfaizah, Imron, M., and Perdanawanti, L. (2017). Algoritma decision tree-j48, k-nearest, dan zero-r pada kinerja akademik. *Seminar Nasional Teknologi Informasi*.

[32] Pereira, R. B., Plastino, A., and Zadrozny, B. (2015). Information gain feature selection for multi-label classication. 6:48–58.

[33] Phyu, T. N. (2009). Survey of classification techniques in data mining. volume 1.

[34] Putra, A. G. and Wirayuda, T. A. B. (2014). Klasifikasi tulisan tangan berupa angka menggunakan random forest dan histogram of oriented gradient. *e-Proceeding of Engineering*, 1(1).

[35] Ranjan, J. (2008). Business justification with business intelligence. *Procedia-Social and Behavioral Sciences*, 38.

[36] Raschka, S. (2017). Naive bayes and text classificaion 1. *arXiv:1410.5329*, 1.

[37] Siyamto, Y. (2017). Pemanfaatan data mining dengan metode clustering untuk evaluasi biaya dokumen ekspor di pt. winstar batam. *Media Informatika Budidarma*, 1(2):28–31.

[38] Song, Y. and Y., L. (2015). Decision tree methods: Applications for classification and prediction. volume 27, Shanghai. Shanghai Archives of Psychiatry.

[39] Surhone, L., Timpledon, M., and Marseken, S. (2010). *Naive Bayes Classifier: Classifier (mathematics), Bayes' Theorem, Probability Theory, Bayesian Inference, Bayesian Probability, Empirical Bayes Method, Statistics, Conditional Probability*. Betascript Publishing.

[40] Turban, E. and dkk. (2005). *Decision Support Systems and Intelligent Systems*. ANDI, Yogyakarta.

[41] Wahyono, T. and Pujiatmoko, L. (2008). *Pengembangan Aplikasi Akuntansi Berbasis Microsoft Virtual Basis.Net*. Elex Media Komputindo, Jakarta.

[42] Wilson, R. A. and Keil, F. C. (2001). *The MIT enclyclopedia of the cognitive sciences*. The MIT Press, London, England.

[43] Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning, Tools and Techniques Ed.3*. Morgan Kaufmann, USA.

[44] Wixom, B. and Watson, H. (2010). The bi-ased orbganization. *IJBIR*.

[45] Zhou, Z., Wang, H., and Lou, P. (2010). *Manufacturing Intelligence for Industrial Engineering: Method for System Self-Organization, Learning and Adaption*. Engineering Science Reference, USA.

BIOGRAFI PENULIS

Herfian Setiawan, lahir di Gorontalo, 19 Oktobre 1991, anak pertama dari dua bersaudara. Penulis menempuh pendidikan formal mulai dari tahun 1998-2002 di SD IMPRES Sukamakmur, Gorontalo, 2002-2006 di SLTP2 MuhamadiahTolangohula, 2006-2009 di SMAN 1 Boliyohuto, 2009-2013 di UniversitasNegeri Gorontalo jurusan Sistem Informasi. Setelah menyelesaikan program S1, penulis bekerja sebagai assiten dosen di jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Gorontalo (UNG).

Kemudian, pada tahun 2014 penulis melanjutkan pendidikan S2 di ITS jurusan Sistem Informasi. Selanjutnya, pada tahun 2015 penulis mengambil program Double Degree antara ITS dan Université Paris Dauphine di Paris, Prancis. Pada tahun 2016 penulis berhasil menyelesaikan studi di Université Paris Dauphine dengan mengambil konsentrasi Business Inteligence. Saat ini penulis bekerja sebagai developer Business Intelligence di perusahaan telekomunikasi di Prancis.